



KERNFORSCHUNGSANLAGE JÜLICH GmbH

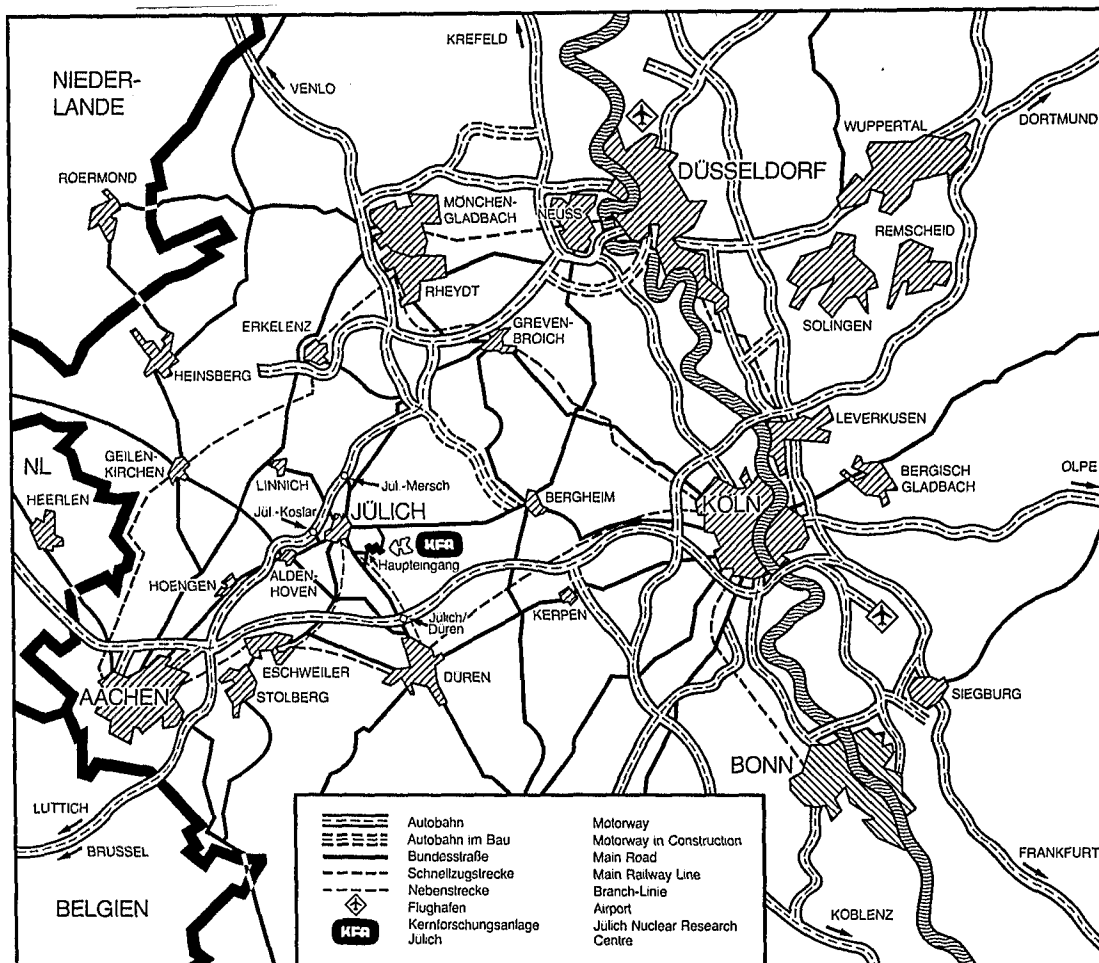
Institut für Kernphysik

**Untersuchung eines
Szintillations-Detektors
besonderer Form**

von

Anton Empl

Jül-Spez-510
Juli 1989
ISSN 0343-7639



Als Manuskript gedruckt

Spezielle Berichte der Kernforschungsanlage Jülich – Nr. 510
 Institut für Kernphysik Jül-Spez-510

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH
 Postfach 19 13 · D-5170 Jülich (Bundesrepublik Deutschland)
 Telefon: 02461/610 · Telex: 833 556-0 kf d

Untersuchung eines Szintillations-Detektors besonderer Form

von

Anton Empl

Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
1.2	Thematik	6
2	OPAQUE (Software)	9
2.1	Programmstruktur	9
2.1.1	Dateneingabe	11
2.1.2	Generieren der Photonen	14
2.1.3	Photonen-Transport	15
2.1.4	Reflexion	16
2.2	Quirl	18
2.2.1	Geometrie	19
2.2.2	Datenfile QUIRL	20
2.2.3	Ergänzende 'Quirl'-Routinen	21
2.2.4	Spezieller Versuchsaufbau	24
2.3	Simulationsergebnisse	24
3	Experiment	37
3.1	Versuchsaufbau	37
3.1.1	Szintillator	38
3.1.2	Lichtleiter (Montage)	41
3.1.3	Photomultiplier	43
3.1.4	Elektronik	46
3.2	Trigger	48
3.2.1	Aufbau	49
3.2.2	Beschreibung	50
3.3	Messung	54
3.3.1	Eichung	56
3.3.2	Meßergebnisse	59

4	Zusammenfassung	71
5	Anhang	75
A	Ergänzung zu den verwendeten PMTs	75
B	Ergänzung zur Elektronik (ACQUIRE)	79
C	Ergänzung zu Kapitel 2	96
D	Simulationsprogramm OPAQUE	101
E	Zusätzliche Ergebnisse der Messung	119

Kapitel 1

Einleitung

1.1 Motivation

Die Suche nach exotischen und speziell gluonischen Zustandsformen der Kernmaterie ist seit der Formulierung der Quantenchromodynamik von fundamentalem Interesse. Die Bedeutung dieser Objekte liegt in der Tatsache, erlaubte und erwartete Zustände im Standard Modell zu sein. Ihre Beobachtung wäre eine aussagekräftige Unterstützung des Quarkmodells.

Um solche Quark-Gluon Zustände zu finden und zu untersuchen, wurde am CERN¹ ein neues experimentelles Programm - **PS 202, JETSET** - gestartet. Allgemeines Ziel ist das Studium seltener oder Zweig-verbotener² Formierungsreaktionen des Typs: $\bar{p}p \rightarrow M_1 M_2$, in denen die neuen Formen der Materie als intermediäre Zwischenzustände am ehesten sichtbar werden sollten.

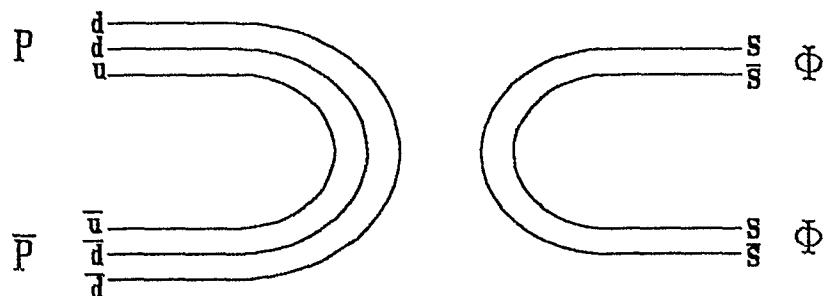


Abbildung 1.1: Quarklinienbild zu der Zweig unterdrückten Reaktion $\bar{p}p \rightarrow \Phi\Phi$.

¹European Organization for Nuclear Research, Genf

²Regel von Zweig: Reaktionen, welche unverbundene Quarklinien aufweisen, werden unterdrückt. [Per82]

Die für eine erste Phase bestimmte Reaktion ist $\bar{p}p \rightarrow K^+ K^- K^+ K^-$ mit besonderem Interesse an dem resonanten Zwischensystem:

$$\bar{p}p \rightarrow \Phi\Phi \rightarrow K^+ K^- K^+ K^-$$

Das Experiment wird am LEAR³ aufgebaut werden. Es besteht im wesentlichen aus einem internen Molekularwasserstoff Target (Cluster Jet), um welches ein kompaktes Detektorsystem angeordnet ist (Abbildung 1.2).

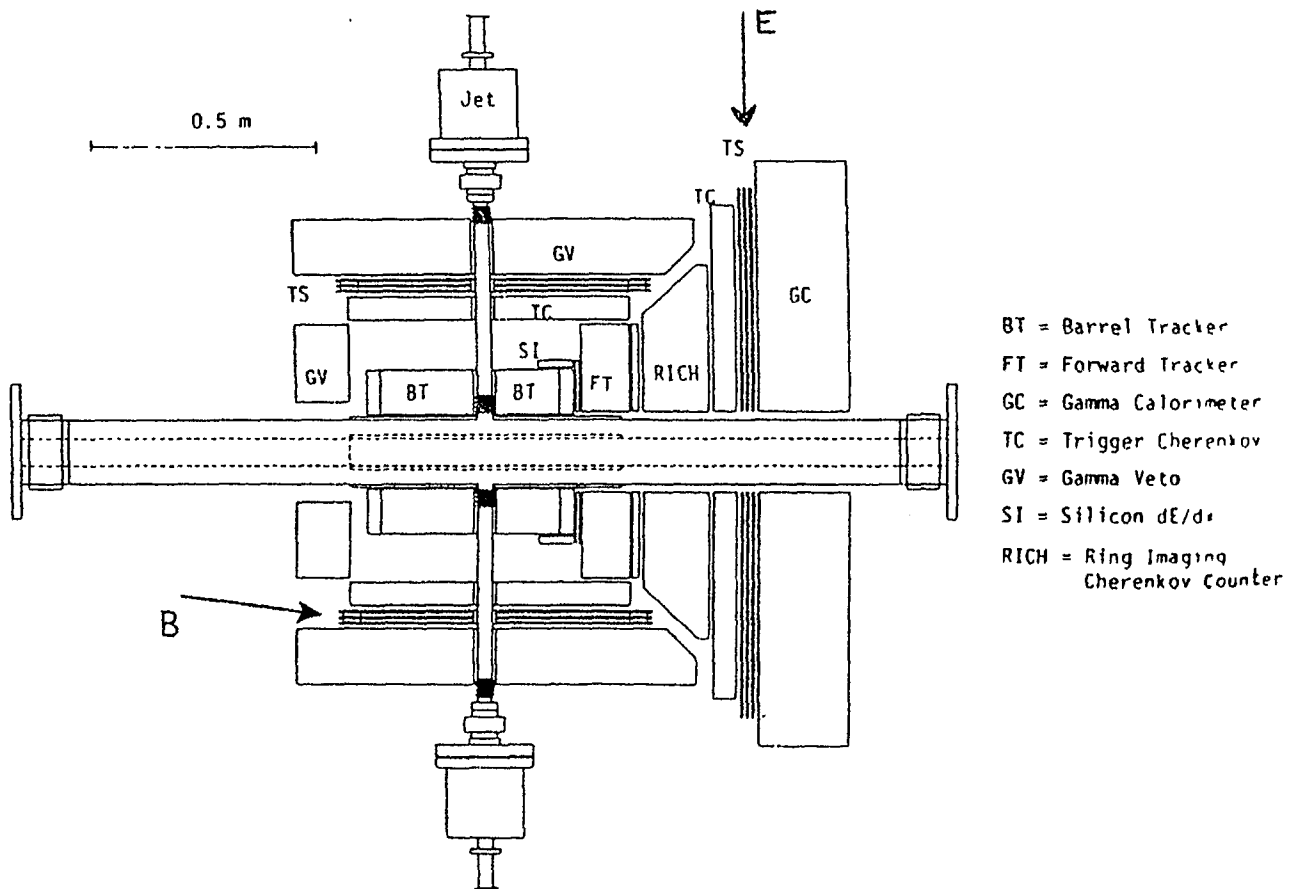


Abbildung 1.2: Das kompakte Detektorsystem JETSET

Die betrachtete Reaktion zeichnet sich durch extrem niedrige Wirkungsquerschnitte aus, so daß von einem sehr ungünstigen Verhältnis zwischen Reaktionsereignissen und Untergrund ausgegangen werden muß. (Abbildung 1.3)

³Low Energie Antiproton Ring

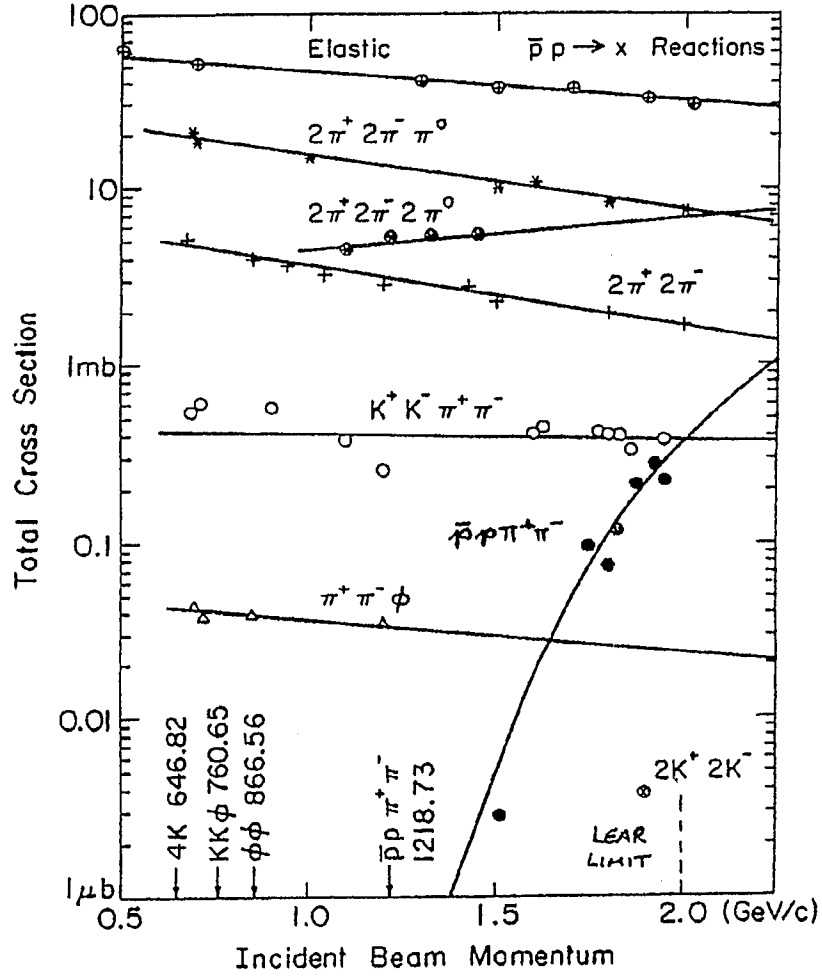


Abbildung 1.3: Totale Wirkungsquerschnitte für die in der ersten Phase zu untersuchende Reaktion $\bar{p}p \rightarrow K^+ K^- K^+ K^-$ (rechts unten) und die dominanten Untergrundreaktionen ($\bar{p}p \rightarrow X$). [Rep88]

Sie erfordern daher ein effektives, *first level* Triggersystem zur Selektion und Datenreduktion. Man nutzt hierbei den Umstand, daß genau 4 geladene Teilchen auftreten (mit einem charakteristischen Pattern) und verlangt diese in einem zweiteiligen Szintillatorhodoskop als Bedingung für ein gutes Event.

Mit Konstruktion, Bau und Einsatz dieses, in beiden Komponenten - Barrel (B), Endcap (E) - dreilagigen Szintillatorhodoskops ist das Institut für Kernphysik (IKP)

der KFA⁴ betraut. Die spezielle Topologie des Hodoskops erlaubt neben der Multiplizitätsbestimmung auch eine schnelle Winkelbestimmung in θ und φ .

[Rep88] [Sef88]

1.2 Thematik

Die Endkappe des schnellen JETSET-Triggers besteht aus einer Lage torstenstückförmiger und zwei Lagen gedrehter Szintillatorsegmente mit entgegengesetztem Drehsinn. Abbildung 1.4 skizziert eine solche gedrehte Lage.

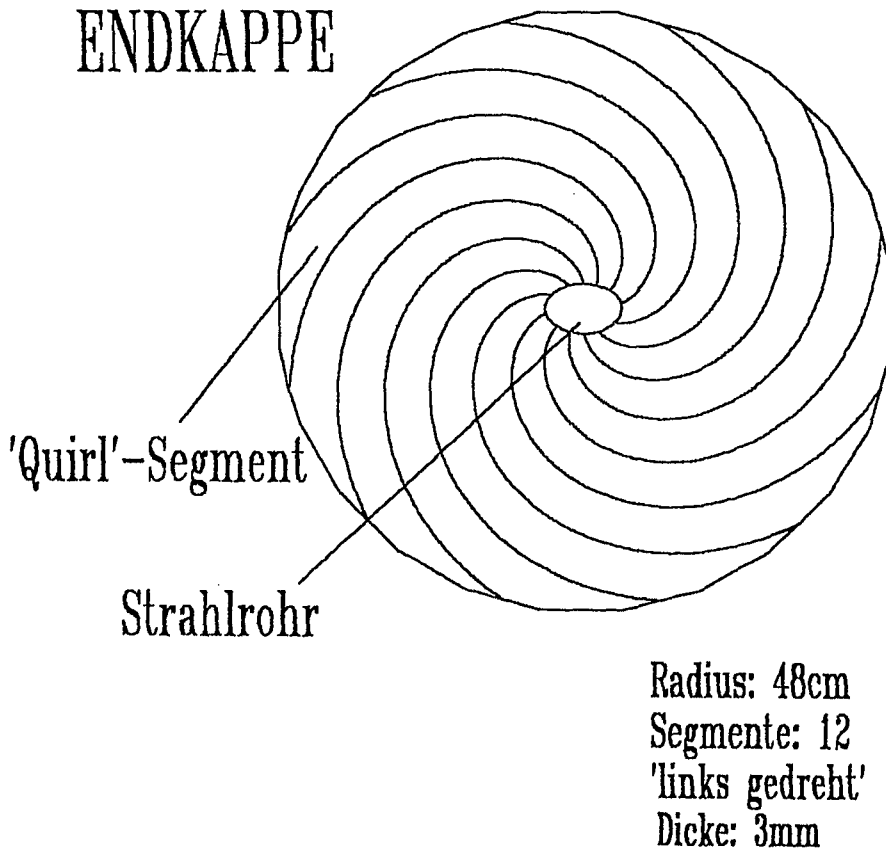


Abbildung 1.4: Gedrehte Szintillatorlage der Endkappe

Gegenstand dieser Arbeit ist die Untersuchung eines gedrehten Szintillatorsegments, besonders im Hinblick auf die Lichtausbeute. Dazu wurde sowohl ein Experiment

⁴Kernforschungsanlage Jülich

mit dem Prototyp eines Segments durchgeführt, als auch ergänzend Software erstellt, die den Photonentransport durch Lichtleiter und Szintillatoren simuliert. Letztere ermöglicht es, ohne Experiment erste Abschätzungen der optischen Eigenschaften für Variationen des 'Quirl'-Segments zu erhalten.

In dem zentralen Kapitel 3 wird der Versuchsaufbau und die Durchführung des Experiments beschrieben. Anschließend werden die Meßergebnisse vorgestellt. Vorher widmet sich jedoch Kapitel 2 der Simulation des Photonentransports (Programmstruktur, Simulationsergebnisse) und der speziellen Geometrie des 'Quirls'. Innerhalb des vierten Kapitels sollen die Ergebnisse der Arbeit kurz zusammengestellt werden.

Der Anhang bietet ergänzende Informationen zu den einzelnen Kapiteln an. Besonders hingewiesen sei auf:

- Herleitung der Wahrscheinlichkeitstransformation
- Listing und kurze Beschreibung des Datenaufnahmeprogramms **Acquire**
- Listing des Simulationsprogramms **OPAQUE**

Kapitel 2

OPAQUE (Software)

Das Programm OPAQUE¹ simuliert den Transport von Photonen in durchsichtigen Körpern. Der Computercode wurde auf einer der Großrechenanlagen der KFA (IBM 3090E) in VS Pascal (Release 2., unter VM/XA) erstellt. Die optionale Graphik benutzt GR-Routinen, eine im ZAM² entwickelte Plotsoftware, die auf dem GKS (graphic kernel system, 7.2 level 2b) basiert.

Es waren spezielle Erweiterungen nötig, um auch Simulationsrechnungen zum 'Quirl' mit seiner besonderen Geometrie durchführen zu können. Auf sie wird im zweiten Teil dieses Kapitels näher eingegangen.

2.1 Programmstruktur

Figur 2.1 gibt das Flußdiagramm zu OPAQUE wieder. In der Initialisierungsphase (*initialize*) werden Informationen zur Geometrie der durchleuchteten Körper, zu den Histogrammen und Laufparameter eingelesen, sowie Datenwerte berechnet, welche sich während der Programmausführung nicht ändern. Die äußere, unbedingte Schleife (*for i:=1 to ...*) umfaßt alle Prozesse, die auf einzelne Photonen angewendet werden. Den Anfang macht das Generieren der 'Photoneigenschaften' Ort, Richtung und Polarisation³ (*generate*).

¹engl. undurchsichtig

²Zentrales Institut für angewandte Mathematik der KFA, siehe KFA-ZAM-0051 GRAF.

³Es wurde mit monochromatischen Photonen gerechnet, deren Frequenz der im Emissionsmaximum des verwendeten Szintillators entsprach. Auf diese Frequenz beziehen sich die Angaben der Hersteller zum Brechungsindex.

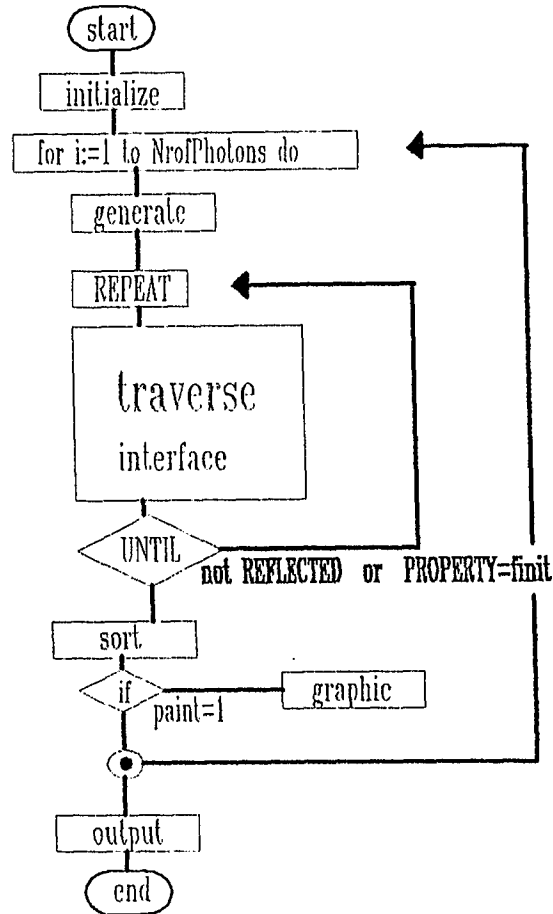


Abbildung 2.1: Flußdiagramm für das Programm OPAQUE.

Innerhalb der bedingten Schleife (*REPEAT ... UNTIL*) leistet die Prozedur 'traverse' den Transport des Lichts durch einen Körper, während die Funktion 'interface' mehrere solcher Körper 'optisch' miteinander verknüpft. Daraufhin werden die vom Benutzer definierten Histogramme gefüllt (*sort*) und wenn gewünscht die Graphik aufgebaut⁴. Nach der durch *NrofPhotons* definierten Anzahl von generierten Photonen endet der Programmablauf mit dem Schreiben der Histogramme in ein File, dessen Name in der Eingabedatei 1 bestimmt wird.

Folgende Files sind zum Ausführen von OPAQUE notwendig⁵:

⁴Sie ist vor allem zur 'visuellen' Kontrolle beim Konstruieren und Zusammensetzen der Körper gedacht.

⁵Zu den Programmen OPAQ EXEC (geschrieben in REXX), OPAQUEF FORTRAN siehe Anhang C.

-----	Datenfile1 mit Geometriedaten und Laufparametern
-----	Datenfile2 mit Daten zu den Histogrammen (Name im Datenfile1 definiert)
OPAQ EXEC	Aufrufen des Programms und Filedefinitionen (Parameter: Name des Datenfiles1)
OPAQUEF FORTRAN	notwendige Initialisierung der GR-Software und der Workstation für das 'graphic kernel system'
OPAQUE PASCAL	Hauptprogramm

Tabelle 2.1: Programmpaket OPAQUE

2.1.1 Dateneingabe

Die Benutzung der beiden Eingabefiles soll anhand von Beispielen aufgezeigt werden. Datenfile 1 ECKE DATA:

```
.....
ecke spec d          datenfile spektren
ecke ergebnis d     ergebnisfile
1 70 20 0            paint, Winkel
200 4                ProfPhotons, Bodies
0.01 0.999           Grenzw., surfaceless
0                   location
1                   shape, Brechungsindex
1.581
1
6 2                  Anzahl der Interfaces
11 1 1 11 1 -1 11 -1 -1 11 -1 1    side, anderer Koerper
10 1 1 10 1 -1 10 -1 -1 10 -1 1
0 0 0 0 0 1         Koordinaten
0 0 0 0 0 1         mode
1                   zweiter Quader
1.581
2
1 1 6 3
10 1 1 10 1 -1 10 -1 -1 10 -1 1
0 1 1 0 1 -1 0 -1 -1 0 -1 1
1 0 0 0 0 1
1                   dritter Quader
1.581
2
3 4 1 2
0 1 1 0 1 -1 -2 -0.9999 -1 -2 -1 -1 1
1 0 1 0 0 0
1                   vierter Quader
1.581
1
1 3
-2 -1 -1 -2 -1 1 0 -1 1 0 -1 -1 -1
-2 -12 -1 -2 -12 1 0 -12 1 0 -12 -1
1 0 0 0 0 2
.....
```

Dieses Format sollte eingehalten werden, das heißt die Anordnung der Daten und auch die Trennungslinien sind fest vorgegeben. Hingegen können rechts von den Daten beliebige Kommentare stehen.

Die ersten beiden Zeilen des Files definieren die Namen der zweiten Eingabe- und der Ausgabedatei. Darauf folgt der Softwarekey *paint*, mit welchem die Grafik ein- und ausgeschaltet werden kann (*paint=1/0*)⁶ und 3 Projektionswinkel für die Darstellung auf dem Bildschirm⁷. Zeile 4 und 5 enthalten der Reihenfolge nach (i) die Anzahl der zu simulierenden Photonen (*NrofPhotons*), (ii) die der durchleuchteten Körper (*bodies*), (iii) die Grenzwahrscheinlichkeit, bis zu der eine Photonspur verfolgt wird und (iv) den Oberflächenverlust. Mit der nächsten Ziffer, ein weiterer Key, wird zwischen homogener und lokaler Erzeugung der Photonen im ersten Körper umgeschaltet (*location=1/0*). Für *location=1* steht in einer weiteren Zeile:

posx, dx, posy, dy .

Für jeden der Körper - im Beispiel sind dies 4 - folgt dann ein Abschnitt mit individuellen Daten:

- Die Form des Körpers wird durch *shape* bestimmt; möglich sind zur Zeit 1=quader und 2=quirl.
- Brechungsindex.
- Anzahl der optisch kontaktierten Flächen; für > 0 stehen in der nächsten Zeile paarweise geordnet, die Nummer der jeweiligen Fläche mit der des angrenzenden Körpers (zur Numerierung der Flächen siehe Anhang C).
- Koordinaten der 8 Eckpunkte (x/y/z) - alle Maße sind konsistent im gesamten Programm in cm angegeben.
- Die Modi der 6 Flächen (*mode*); es bedeuten
 0=*passiv*: Grenzfläche zur Luft
 1=*aktiv* : Grenzfläche zu anderem Körper
 2=*fnit* : Fläche, die der Auskopplung dient

In Abbildung 2.2 ist für ein 'Event' die Grafik zu der in ECKE DATA definierten Kombination von Körpern mit der angegebenen Projektion dargestellt. Die dreieckige Grundfläche des dritten Quaders⁸ erhält man durch Zusammenziehen zweier Eckpunkte (Abstand $\frac{1}{1000}$ cm).

⁶Kursiv gedruckt sind im Weiteren die Namen von Programmvariablen und Unterprogrammen.

⁷(Eulerwinkel), Defaultwerte: x waagrecht - y senkrecht - z in den Bildschirm hinein.

⁸Körper die 8 Ecken aber beliebige Form besitzen, sollen vereinfachend als 'Quader' bezeichnet werden.

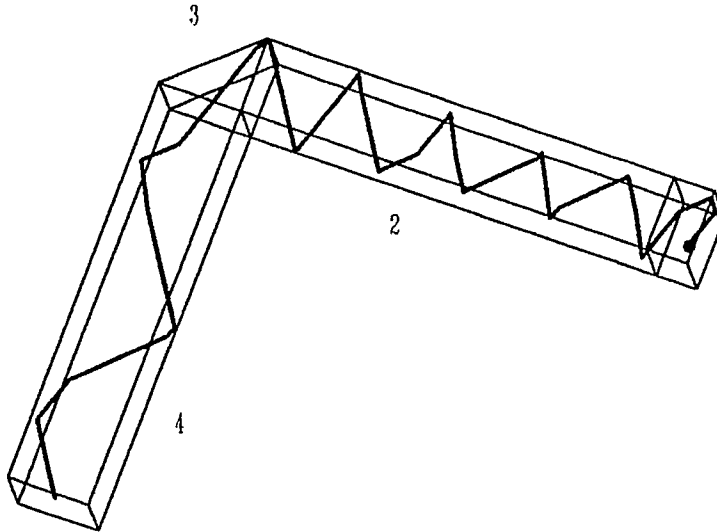


Abbildung 2.2: Projektion der durch ECKE DATA erzeugten Anordnung von Lichtleitern. Der Punkt rechts zeigt den Ort der Generation, die Stirnfläche von Quader 4 dient der Lichtauskopplung.

Durch die zweite Eingabedatei wird die Form der Histogramme bestimmt, die während der Programmausführung gefüllt werden sollen. Datenfile 2 OPAQ SPEC:⁹

2				anzahl der spektren	1
angular distribution				titel	1
-2.05	2.05	205	x: min max bins		
-2.05	2.05	205	y: min max bins		
efficiency					2
-79.5	43.5	205			
-51.5	71.5	205			

Die erste Zahl gibt an wieviele Spektren man definieren möchte und für jedes von diesen folgen 3 Zeilen mit Titel und pro Koordinate Minimum, Maximum, Anzahl der Bins.

Trifft ein Photon auf eine Fläche, die der Lichtauskopplung dient (*mode=finit*), so wird die Prozedur *sort* ausgeführt, in welcher die interessierenden Werte in die Spektren einsortiert werden.

⁹Es wurde dazu eine zweite Datei angelegt, um feste 'Sätze' von Parametern leicht für verschiedene Lichtleiterkombinationen verfügbar zu machen. Um ein eindimensionales Histogramm zu erhalten, setzt man für die x-Koordinate: 0 1 1 und die Grenzen für die y-Koordinate wie gewünscht.

2.1.2 Generieren der Photonen

Zu Beginn eines 'Events' erzeugt das Programm 3 Eigenschaften der Photonen.

(a) Ort (*point*)

Hierfür sind zwei unterschiedliche Verfahren implementiert, die über die Dateneingabe angewählt werden. Einmal kann man einen Punkt (x/y) vorgeben ($location=1$), so daß der Startort der Photonen in einem Bereich $(x + \Delta x/y + \Delta y)$ mit zufälliger Höhe $z_{min} < z < z_{max}$ liegt (Prozedur *Generate_Photon*). Dies entspricht ungefähr der späteren experimentellen Situation¹⁰. Auf der anderen Seite besteht die Möglichkeit jeden beliebigen, achteckigen Körper (default: der zuerst im Datenfile 1 definierte) 'homogen zu beleuchten' ($location=0$), das heißt die Wahrscheinlichkeit für die Erzeugung eines Photons in einem Raumpunkt des Körpers ist überall gleich groß (Prozedur *Generate_Quader*).

(b) Richtung (*aim*)

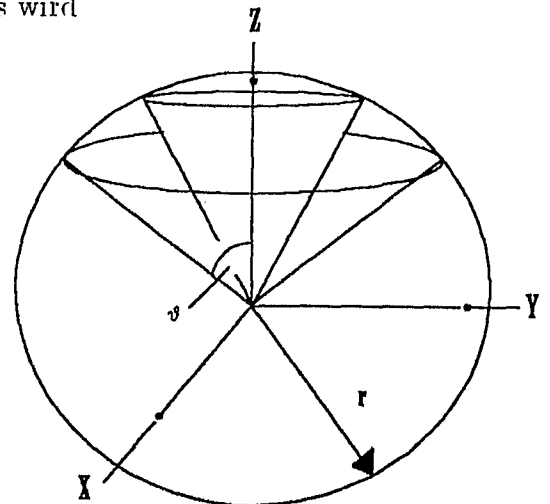
Da das Szintillationslicht keine Korrelation mit der Impulsrichtung der energiedeponierenden Teilchen aufweist, wird die Richtung der Photonen im Raum isotrop 'monte carliert'.

Um einen Vektor mit im Raum isotroper Richtungsverteilung zu erhalten, geht man zweckmäßig von Kugelkoordinaten aus¹¹. Der Radius wird konstant gehalten und der Polarwinkel φ gleichverteilt angenommen:

$$r = const (= 1), \quad \varphi = 2\pi \cdot rndm$$

Hierbei ist *rndm* eine auf dem Intervall $[0,1]$ gleichverteilte Zufallszahl¹².

Die Spitze eines isotrop im Raum gewürfelten Vektors konstanter Länge, bestreicht eine Kugeloberfläche homogen. Damit folgt für Kreisringe auf dieser Kugel (siehe Skizze nebenan):



¹⁰In Sintillator entstehen die Photonen entlang der Teilchenspuren. Die Elektronen wurden im Mittel in z-Richtung eingestrahlt.

¹¹Auf einem Würfel x, y, z gleichverteilt anzunehmen reicht nicht aus! Dabei werden die Raumwinkel in Eckennähe überproportional belegt.

¹²Der Zufallszahlengenerator RNDM wurde der CERN-Programmbibliothek (Kernlib) entnommen.

$$\begin{aligned}
D = \text{const} &= \frac{N}{U} \\
U = 2\pi r' &= 2\pi r \sin(\vartheta) \\
\Rightarrow N &= D 2\pi r \sin(\vartheta) \quad \Rightarrow N \sim \sin(\vartheta)
\end{aligned}$$

D : Punktedichte, N : Anzahl der Punkte, U : Umfang des Kreisrings

N kann auch als Häufigkeit betrachtet werden, mit der ein jeweiliger Kreisring von der Vektorspitze bestrichen wird. Dies aber bedeutet, daß man für den azimutalen Winkel eine sinusförmige Wahrscheinlichkeitsverteilung benötigt. Sie wird durch folgende Beziehung¹³ erzeugt:

$$\vartheta = \arccos(1 - 2 \cdot \text{rndn})$$

(c) Polarisation (*polaris*)

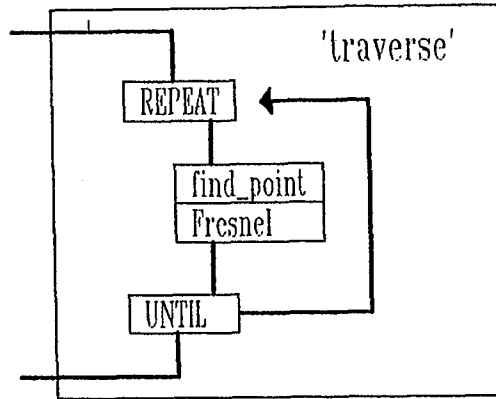
Die Wahrscheinlichkeit der Reflexion von Licht an einer Grenzfläche hängt von seiner Polarisation ab (siehe Fresnel-Reflexion, 2.1.4). Daher wird ein zufälliger Vektor *polaris* in der Ebene senkrecht zur Photonrichtung erstellt, der das Verhältnis paralleler zu senkrechter Polarisation bestimmt.

2.1.3 Photonen-Transport

Die Routine *traverse* simuliert den Transport der Photonen durch einen Körper. Sie besteht im wesentlichen aus einer bedingten Schleife, für die es drei Abbruchbedingungen gibt:

- das Photon trifft auf eine Fläche mit optischen Kontakt zu einem anderen Körper (*mode=activ*)
- es trifft auf eine Fläche, die der Lichtauskopplung dient (*mode=finit*)
- das Produkt der Wahrscheinlichkeiten für die Reflexionen ist kleiner als die durch die Variable *GrenzWahrscheinlichkeit* geforderte Mindestwahrscheinlichkeit (z.B.: $\frac{1}{100}$)

¹³Die Verschiebung des Arguments resultiert aus dem Wertebereich $[-1,1]$ des Cosinus für $0 \leq \vartheta \leq \pi$. Zur Herleitung siehe Anhang C.



Eingabeparameter, der für jede Körperform spezifischen Unterprogramme *find_point*, sind lediglich Ort und Richtung (*point*, *aim*) der Photonen. Sie berechnen den Reflexionspunkt (*refl*) und die Ebene an der reflektiert wird (*plane*), welche wiederum Inputwerte der Prozedur *Fresnel* sind¹⁴.

Hier kommt der modulare Aufbau des Programms zum Ausdruck. Um weitere Formen von Lichtleitern zu erlauben, braucht man nur die speziellen Unterprogramme *find_point*, *graphic_* und *generate_* zu ergänzen, wie dies analog für den Quirl geschehen ist¹⁵.

Den Übergang zu angrenzenden Körpern, in beiden Richtungen, stellt die Funktion *interface* her. Sie erlaubt nur vollständige Seitenflächen zu verbinden, partielle optische Kontakte müssen gesondert programmiert werden. Ein Beispiel dafür liefert die Routine *Plexi* (siehe 2.2.4).

Der Dämpfung in den Lichtleitern wird durch die Routine *Attenuate* Rechnung getragen.

2.1.4 Reflexion

Die Routine *Fresnel* beinhaltet die physikalischen Kenntnisse über den Prozess der Reflexion beziehungsweise Transmission.

Fällt Licht auf die Grenzfläche zweier Medien mit unterschiedlichen Brechungsindizes (n_1, n_2), so berechnet sich das Verhältnis von reflektierter (r) (gebrochener (t)) zu einfallender (i) Amplitude nach den **Fresnel Formeln**¹⁶. [Alo70] Für die parallel

¹⁴Für die beiden unterstützten Formen heißen die Routinen *point_quader* und *point_spirale*. Auf letztere wird unter 2.2 näher eingegangen. Der Algorithmus *point_quader* berechnet die Schnittpunkte der Körperflächen mit der durch *point* und *aim* gegebenen Geraden.

¹⁵Verzichtet man auf die Grafik und generiert die Photonen lokal, so muß nur das Unterprogramm *find_point* hinzugefügt werden.

¹⁶Eine exakte Herleitung aus den Maxwell-Gleichungen findet sich in [Jac83]

(π) und senkrecht (σ) zur Einfallsebene polarisierten Komponenten des elektrischen Feldes erhält man:

$$\begin{aligned} r_\pi &= \frac{E_{r,\pi}}{E_{i,\pi}} = \frac{\tan(\alpha_i - \alpha_t)}{\tan(\alpha_i + \alpha_t)} \\ r_\sigma &= \frac{E_{r,\sigma}}{E_{i,\sigma}} = \frac{\sin(\alpha_t - \alpha_i)}{\sin(\alpha_t + \alpha_i)} \\ t_\pi &= \frac{E_{t,\pi}}{E_{i,\pi}} = \frac{2\sin(\alpha_t)\cos(\alpha_i)}{\sin(\alpha_i + \alpha_t)} \\ t_\sigma &= \frac{E_{t,\sigma}}{E_{i,\sigma}} = \frac{2\sin(\alpha_t)\cos(\alpha_i)}{\sin(\alpha_i + \alpha_t)\cos(\alpha_i - \alpha_t)} \end{aligned}$$

Die in den Gleichungen auftretenden Winkel α_i, α_t sind durch das Snellius'sche Brechungsgesetz

$$\frac{\sin(\alpha_i)}{\sin(\alpha_t)} = \frac{n_2}{n_1}$$

miteinander verknüpft.

Zur Berechnung der Reflexionswahrscheinlichkeit wird zunächst der zu Beginn eines 'Events' zufällig generierte, auf 1 normierte Vektor *polaris*, der die Schwingungsebene des Lichts senkrecht zur Ausbreitungsrichtung bestimmt, in seine Komponenten parallel und senkrecht zur Einfallsebene zerlegt.

$$\begin{aligned} |\vec{P}_{polaris,a}|^2 &= 1 & (\text{am Anfang}) \\ \vec{P}_{polaris} &= E_{i,\sigma} + E_{i,\pi} \end{aligned}$$

Mit den obigen Koeffizienten, unter Berücksichtigung der neuen Richtung von E_π , ergibt sich für den Polarisationsvektor nach der Reflexion (Transmission):

$$\begin{aligned} \vec{P}_{polaris} &= r_\sigma E_{i,\sigma} + r_\pi E'_{i,\pi} \\ (\vec{P}_{polaris} &= t_\sigma E_{i,\sigma} + t_\pi E'_{i,\pi}) \end{aligned}$$

Das Amplitudenquadrat von *polaris* ist ein Maß für die Intensität des reflektierten (gebrochenen) Lichts. Mit der gewählten Normierung stellt $|\vec{P}_{polaris}|^2$ gerade die

gesuchte Wahrscheinlichkeit der Reflexion (Transmission) eines Photons dar. Dieses Verfahren wird iterativ pro Grenzfläche auf den Vektor *polaris* angewendet.

Die Routine *fresnel* berechnet auch den neuen Richtungsvektor *aim*. Im Fall der Reflexion erhält man ihn einfach durch Verbinden des an der Reflexionsebene gespiegelten Punkts *point* (\rightarrow *anti*) mit dem Reflexionspunkt *refl*. Innerhalb der gesamten Routine werden keine trigonometrischen Funktionen verwendet, so daß für die Transmission die neue Richtung aus dem Verhältnis *An* zu *An'* mit

$$An' = \sqrt{\left(\frac{n_2}{n_1} Hy\right)^2 - Ge^2}$$

berechnet wird¹⁷.

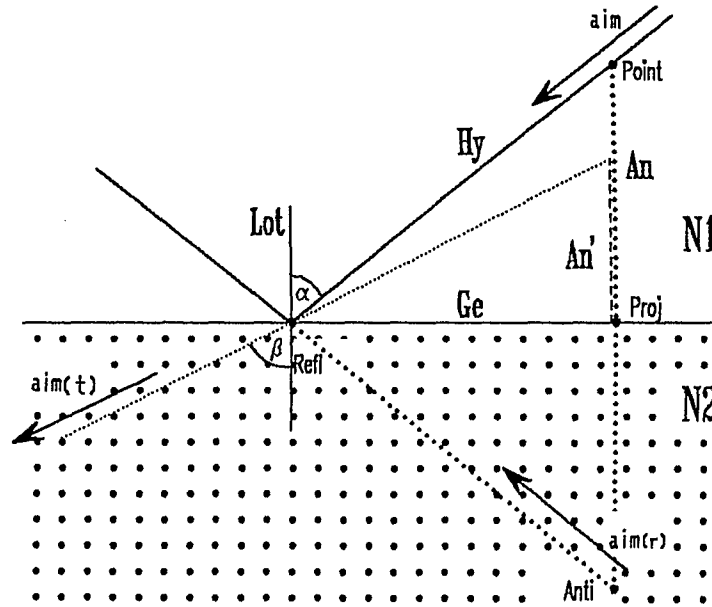


Abbildung 2.3: Konstruktion der neuen Photonrichtung. Für die Reflexion erhält man sie durch spiegeln von *point* an der Reflexionsebene \rightarrow *aim(r)*; im Falle der Transmission über das Verhältnis $An'/An \rightarrow$ *aim(t)*.

2.2 Quirl

In dem hier folgenden Unterkapitel 2.2 soll auf die Geometrie des 'Quirls' und die für das Programm notwendigen Erweiterungen eingegangen werden.

¹⁷ *An, Ge, Hy* bedeuten hierbei Ankathete, Gegenkathete und Hypothenuse.

2.2.1 Geometrie

Die Randkurven der gedrehten Szintillatorsegmente heißen
Archimedische Spiralen:

'Die Kurven sind die Bahn eines Punktes, der mit konstanter Geschwindigkeit v längs eines sich mit konstanter Winkelgeschwindigkeit ω , um seinen Anfangspunkt drehenden Stabes gleitet.

*Gleichung in Polarkoordinaten: $\rho = a|\varphi|$, $a = v/\omega \geq 0$,
 $-\infty \leq \varphi \leq \infty$ '*

[BS84]

Der Radius nimmt also proportional mit dem Drehwinkel zu. Im weiteren wird die Darstellung in karthesischen Koordinaten gewählt:

$$x = \frac{r_{\max} \varphi}{\varphi_{\max}} \cos(\varphi + \varphi_o + \Delta)$$

$$y = \frac{r_{\max} \varphi}{\varphi_{\max}} \sin(\varphi + \varphi_o + \Delta)$$

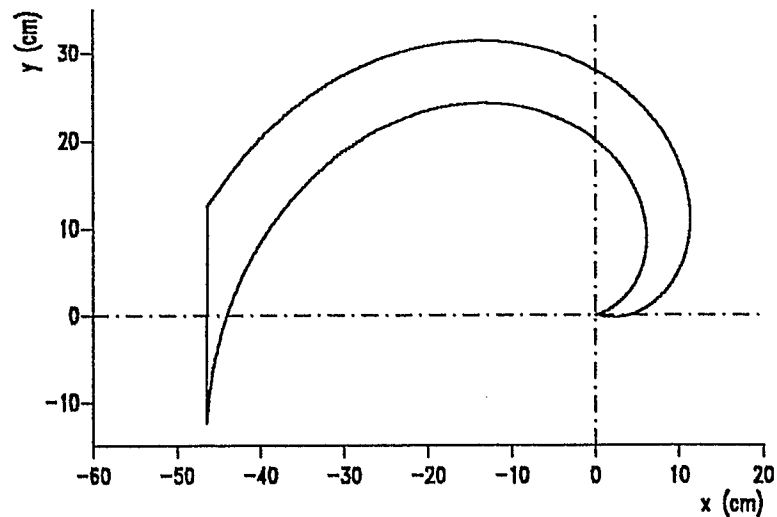


Abbildung 2.4: Projektion der Geometrie des 'Quirl'-segments ohne den angesetzten Quader ($x_{Min} = -46.36cm$).

φ_{max} bedeutet den Maximalwert des Winkels φ : $0 \leq \varphi \leq \varphi_{max}$. Δ ist je nachdem, ob innere oder äußere Randkurve, 0° oder gleich dem Öffnungswinkel zwischen den beide Spiralen (in einer Ebene). Über den Winkel φ_o kann das gesamte 'Quirl'-segment in die gewünschte Lage gedreht werden. r_{max} bezeichnet den maximalen Radius. Im betrachteten Fall waren:

- $\varphi_{max} = 180^\circ$
- Öffnungswinkel = 30° , $\rightarrow \Delta = 0^\circ$ oder $\Delta = 30^\circ$
- $\varphi_o = -15^\circ$, damit wird erreicht, daß die Stirnfläche in der xy-Ebene liegt.
- $r_{max} = 48cm$
- Dicke $z = 3mm$

Die Stirnfläche entsteht durch verbinden der Endpunkte der Spiralen. Um die Form des im Experiment benutzten Prototyps zu erzeugen, wird ein Quader mit der gewünschten Form angesetzt.

2.2.2 Datenfile QUIRL

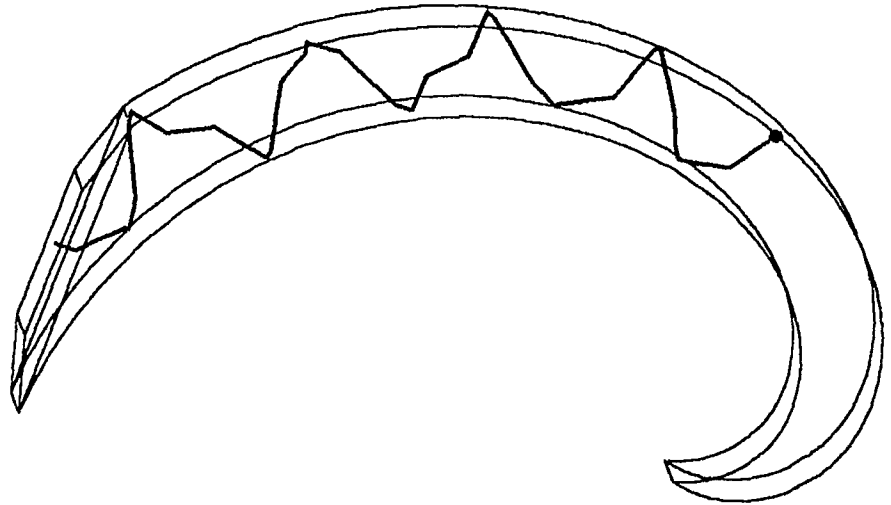


Abbildung 2.5: Projektion der im Datenfile QUIRL definierten Lichtleiterkombination ('Quirl' + angesetzter Quader).

In Abbildung 2.5 ist die Projektion eines Events im 'Quirl' wiedergegeben. Der Punkt bezeichnet den Ort der Generation. Für dieses spezielle Event erreicht das Photon eine der Flächen, die der Auskopplung dienen. Das dazugehörige Datenfile QUIRL DATA hat die Form:

```

.....
quir1 spec d                               datenfile histogr.
quir1 ergebnis d                           ergebnisfile
2 70 20 0                                  paint, winkel
100                                         nroffphotens, bodies
0.01 1                                     grenzw., surfaceless
0                                           location
.....
2                                           shape : 'Quirl'
1                                           Modus der Stirnflaeche (activ)
1.581                                       Brechungsindex
180 30 48 3.0                             phi, delta, radius, dicke
.....
1
1.581
1
1 1
-46.364439 -12.4233 0.0 -46.364439 12.4233 0.0
-46.364439 12.4233 3.0 -46.364439 -12.4233 3.0
-48.0000 -6.3193 0.0 -48.0000 6.3193 0.0
-48.0000 6.3193 3.0 -48.0000 -6.3193 3.0
.....1 2 0 2 0 2.....

```

Der untere Teil definiert den Quader. Für den 'Quirl' wird als *shape* der Wert 2 eingegeben und nach dem Brechungsindex folgen der Reihe nach: maximaler Winkel φ , Öffnungswinkel, maximaler Radius und Dicke¹⁸.

2.2.3 Ergänzende 'Quirl'-Routinen

Es sind dies im Prinzip 3 Unterprogramme:

- (i) *graphic_spirale*
- (ii) *generate_spirale*
- (iii) *point_spirale*

Auf die Graphik soll nicht weiter eingegangen werden.

- (ii) *generate_spirale*

Mit dieser Routine werden im 'Quirl' homogen verteilt Photonen generiert. Um das Verfahren möglichst effektiv zu gestalten, 'monte carliert' man zunächst einen Punkt auf einer Kreisscheibe mit Radius r_{max} . Damit diese Punkte gleichmäßig verteilt sind, nimmt man in Polarkoordinaten φ als Gleichverteilung an:

$$\varphi = 2\pi \text{rndm}$$

In Analogie zu Kapitel 2.1.2 denkt man sich den Kreis in Kreisringe zerschnitten und erhält (U=Umfang, N=Anzahl der Punkte):

¹⁸Die Dicke wurde der Anschaulichkeit halber für die Abbildung 2.5 von 3mm auf 3cm heraufgesetzt. In der Graphik des 'Quirls' fehlt der Einfachheit halber die für die Rechnung vorhandene, zylindrische Seitenfläche im Zentrum.

$$\begin{aligned}
 D = \text{const} &= \frac{N}{U}, U = 2\pi r \\
 \Rightarrow N &= DU \\
 \Rightarrow N &\sim r
 \end{aligned}$$

Mit der in Anhang C hergeleiteten Transformation für Wahrscheinlichkeitsverteilungen folgt:

$$\begin{aligned}
 f(y) &= cy, r = F^{-1}(y) \\
 \Rightarrow y(r) &= c'\sqrt{r}
 \end{aligned}$$

r stellt dabei die gleichverteilte Zufallszahl dar, von der ausgegangen wird:
 $r = rndm \quad (c' = r_{max})$.

Der Kreis wird dann auf einen Kreissektor (Öffnungswinkel= Öffnungswinkel der Spiralen) gefaltet, in die Spiralforn gedreht und die sich ergebende kreisförmige Stirnfläche abgeschnitten. Zum Simulieren der exakten Stirnfläche schaltet man zwischen den Generatoren *generate_quader* und *generate_spirale* im Verhältnis der Volumina ($V_q = 5.06\%$, $V_s = 94.94\%$) um.

(iii) *point_spirale*

Der 'Quirl' zeichnet sich durch 3 verschiedenartige, scitliche Begrenzungsflächen aus. Für jede existiert eine eigene Unterroutine, welche Reflexionspunkt und -ebene berechnet.

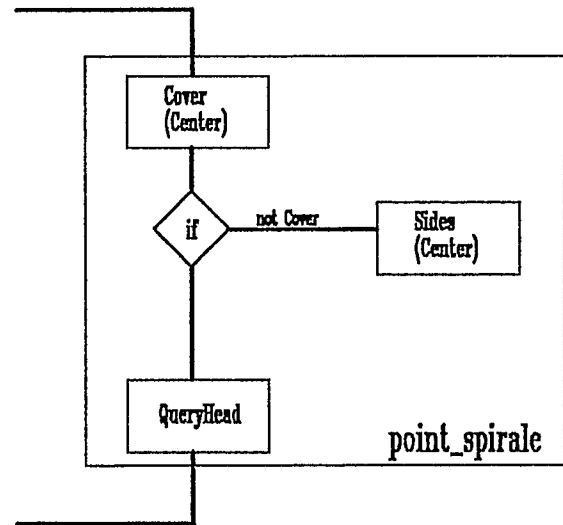
- Center*: zylindrische Fläche im Zentrum, durch das Strahlrohr¹⁸ bedingt.
- QueryHead*: rechteckige Stirnfläche in der yz-Ebene mit der Bedingung¹⁹ $x = xMin$.
- Sides*: gekrümmte Seitenflächen entlang der Spiralen.

Dazu kommt noch die als Funktion realisierte Routine *Cover*. Sie berechnet den gesuchten Reflexionspunkt für 'Boden-' und 'Deckel-' Fläche und wird, da sie weniger rechenintensiv ist als die Prozedur *Sides*, immer zuerst ausgeführt; auch ist die Wahrscheinlichkeit einer Reflexion hier am größten. Nur wenn das Ergebnis von *Cover*

¹⁸Das elliptische Strahlrohr ($a = 6.8cm / b = 3.0cm$) wird in erster Näherung als kreisförmig angenommen.

¹⁹Diese Bedingung ergibt sich aus der speziellen Lage des 'Quirls' im Koordinatensystem.

negativ ausfällt, werden die Seitenflächen in der Routine *Sides* untersucht. Innerhalb von *Cover* und *Sides* findet, wenn nötig die Reflexion an der zylindrischen Fläche im Zentrum statt (*Center*). Am Ende von *point_spirale* wird dann jeweils getestet, ob die Stirnfläche getroffen wurde.



Es soll hier lediglich kurz auf die Routine *Sides* näher eingegangen werden: Das Auffinden der Reflexionsorte besteht mathematisch in der Berechnung des Schnittpunktes einer Geraden (gegeben durch *point* und *aim*) mit den spiralförmigen Seitenflächen. An dieser Stelle reicht es, die Projektion in der *xy*-Ebene zu betrachten.

$$point_x + t * aim_x = r_{max} \frac{\varphi}{\varphi_{max}} \cos(\varphi + \Delta)$$

$$point_y + t * aim_y = r_{max} \frac{\varphi}{\varphi_{max}} \sin(\varphi + \Delta)$$

Dies sind zwei Gleichungen mit zwei Unbekannten (t, φ), die sich somit eindeutig bestimmen lassen. Leider führt das System aber auf eine transzendente Gleichung und man muß die Lösungen näherungsweise z.B. mit dem Newtonverfahren berechnen. Eine weitere Schwierigkeit ergibt sich aus der Tatsache, daß es auch Geraden mit 2 Schnittpunkten gibt²¹.

Die Routine *newton* bestimmt die Schnittpunkte und zwar nacheinander für die innere und äußere Spirale. Mit dem kleinsten, positiven t werden dann der Reflexionspunkt *refl* und die Reflexionsebene *plane* berechnet.

²¹Solange der Winkel φ nur bis 180° läuft sind es maximal 2 Schnittpunkte.

2.2.4 Spezieller Versuchsaufbau

Um einen Vergleich zwischen Experiment und Simulationsrechnungen zu ermöglichen, mußten die Plexiglaslichtleiter in den geometrischen Datensatz aufgenommen werden. Das Datenfile ist in Anhang C wiedergegeben. Die Lichtleiter wurden gleichmäßig über die Stirnfläche verteilt.

Zusätzlich bedurfte es der Routine *Plexi*, da jetzt die Seite eines Körpers weder vollständig noch von einem einzelnen Körper kontaktiert war. Diese Prozedur entscheidet, ob ein Photon zwischen den Ansatzflächen zweier Lichtleiter reflektiert wird (a) oder bestimmt welcher Lichtleiter den Weitertransport übernimmt (b).

Der Aufruf erfolgt aus dem Unterprogramm *ponit_quader* heraus.

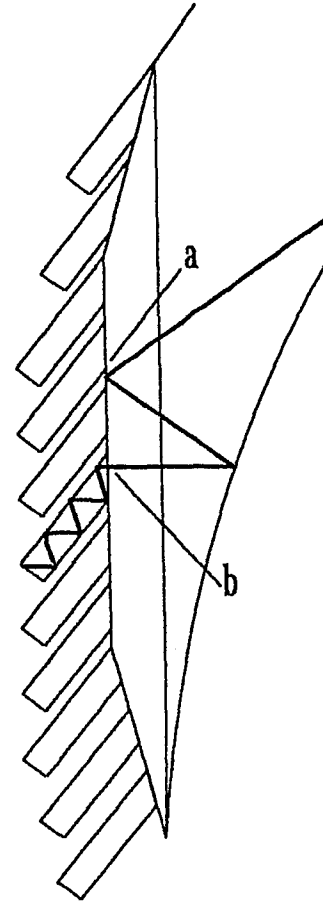


Abbildung 2.6: Dargestellt ist die Stirnfläche mit den angesetzten Plexiglaslichtleitern.

2.3 Simulationsergebnisse

- (a) Zunächst sollen Ergebnisse referiert werden, die mit homogen im gesamten 'Quirl'-Segment generierten Photonen, erzielt wurden. Die Zahl der simulierten Photonen betrug 300 000.

(i) Effizienz

Die Effizienz stellt den über die Impulskoordinaten integrierten Phasenraum für die Photonen im 'Quirl' dar²². Um sie rechnerisch zu bestimmen, wird über das 'Quirl'-Segment in der xy -Ebene ein zweidimensionales Raster (Rasterung $\Delta x, \Delta y$) gelegt. Erreicht ein am Ort (x/y) erzeugtes Photon die Stirnfläche mit der Wahrscheinlichkeit p , so wird diese in das 'bin' $(x \pm \Delta x/y \pm \Delta y)$ einsortiert.

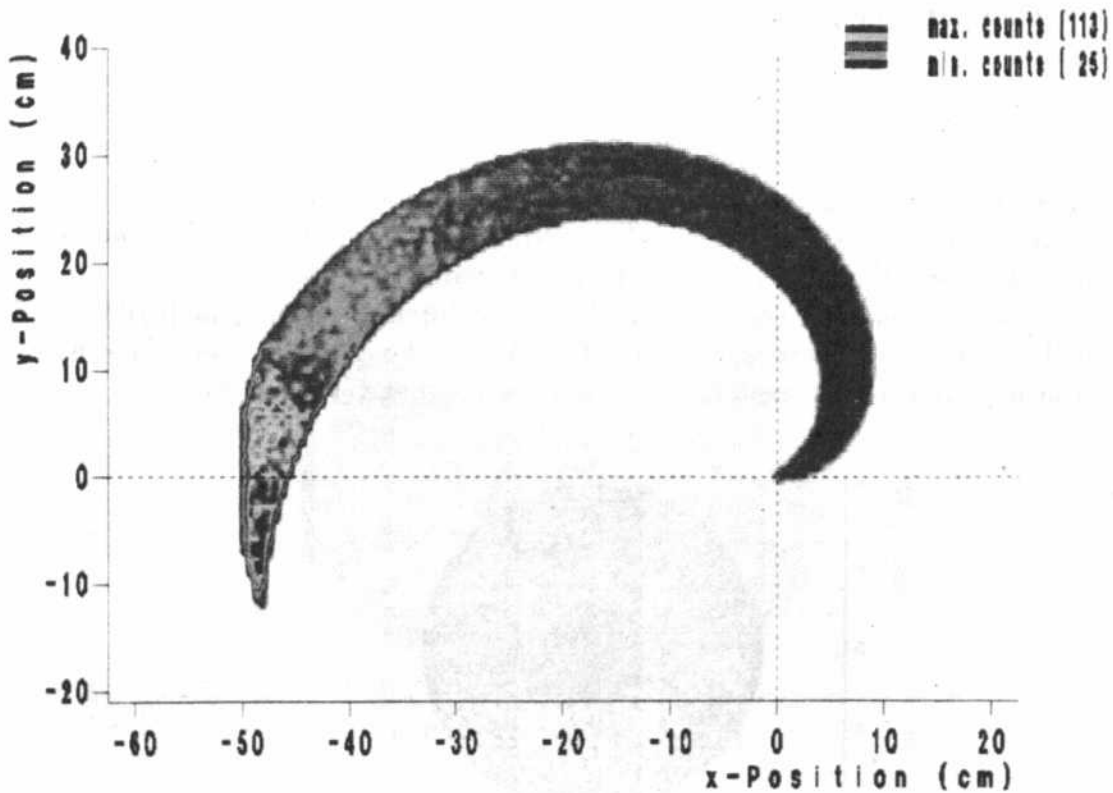


Abbildung 2.7: Darstellung der Effizienz eines 'Quirl'-segments.

Man kann an dem so erstellten Diagramm 2.7 (die Summe der Wahrscheinlichkeit p pro 'bin' wurde in einen Farbcode umgesetzt: die Skala reicht von dunkel blau (25) bis rot (113)) die relative Wahrscheinlichkeit ablesen, mit der ein am Ort (x/y) generiertes Photon zur Stirnfläche gelangt und im Photomultiplier, unter Voraussetzung idealer Lichtleitung, nachgewiesen werden kann.

²²Der Impulsbetrag ist konstant und durch die Frequenz der monochromatisch angenommenen Photonen gegeben. Da die z -Koordinate aus Symmetriegründen keine weiteren Informationen enthält, wird auch über sie integriert.

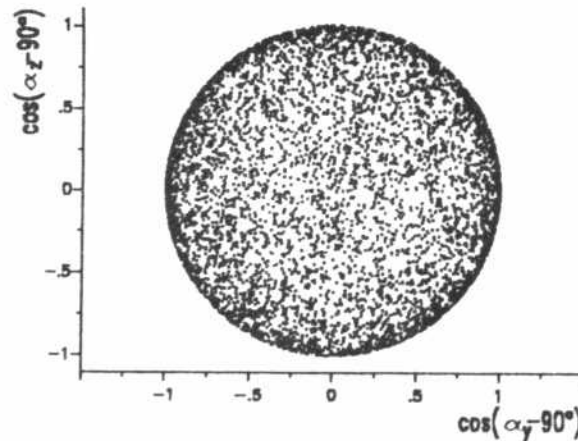
Das Diagramm läßt sich in zwei Bereiche unterteilen:

- Vom Zentrum aus ändert sich die Effizienz auf einer weiten Strecke längs des 'Quirls' kaum²³. Dieser homogene Bereich soll als 'Plateau' bezeichnet werden.
- In der direkten Nähe der Stirnfläche nimmt die Effizienz stark zu²⁴. Der Grund hierfür liegt neben den kurzen Wegstrecken (\rightarrow geringe Dämpfung) und wenigen Reflexionen hauptsächlich im Anwachsen des Raumwinkels, für den Photonen überhaupt die Stirnfläche erreichen können.

(ii) Winkelverteilung

Die Winkelverteilung der Photonen an der Stirnfläche ist für die Ankopplung der Lichtleiter von Interesse. Um diese Winkelverteilung darzustellen, werden die Richtungscosinus²⁵ - $\cos(\alpha_y)$, $\cos(\alpha_z)$ - gegeneinander aufgetragen.

Für isotrop generierte, normierte Richtungsvektoren (Ursprung in (0/0/0)) entspricht diese Darstellung gerade der Projektion, der homogen auf einer Kugeloberfläche verteilten Durchstoßpunkte, auf die yz-Ebene (Figur 2.8).



Abbildungung 2.8: Projektion der homogen bestrichenen Einheitskugel-Oberfläche. Die Punktdichte nimmt vom Zentrum nach außen hin sinusförmig zu.

²³Im Zentrum beträgt die relative Effizienz $25 \frac{\text{Photonen}}{\text{bin}}$ und bei den Anfängen der hellblauen Region etwa $35 \frac{\text{Photonen}}{\text{bin}}$.

²⁴Maximum $113 \frac{\text{Photonen}}{\text{bin}}$, grün bedeutet etwa $50 \frac{\text{Photonen}}{\text{bin}}$

²⁵Das sind die Cosinus der Winkel zwischen der Photonrichtung und ihren Projektionen auf die y- und z-Koordinatenachsen. Die Stirnfläche liegt in einer von diesen beiden Achsen aufgespannten Ebene.

In Abbildung 2.9 ist die Winkelverteilung für die homogen im 'Quirl' generierten Photonen an der Stirnfläche gezeigt²⁶. Auch hier ist die Information in einen Farbcode umgesetzt²⁷. Man erkennt deutliche Abweichungen von der in Figur 2.8 dargestellten, isotropen Winkelverteilung, die durch die spezielle Form des durchleuchteten Körpers bedingt sind. Das Kreuz in der Mitte repräsentiert die Richtung senkrecht zur Stirnfläche (antiparallel zur x-Achse).

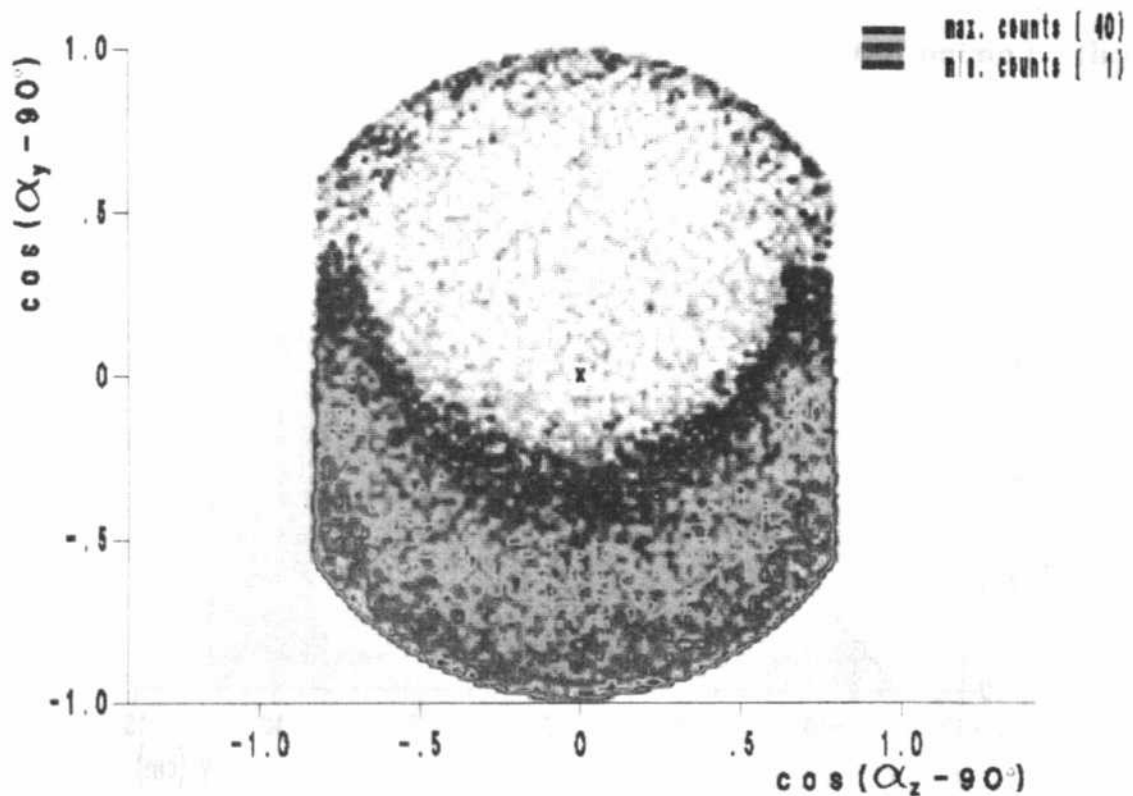


Abbildung 2.9: Winkelverteilung der Photonen, die die Stirnfläche des 'Quirls' erreichen.

- Die Winkel sind in z-Richtung eingeschränkt.

Der 'Quirl' besteht aus zwei planparallelen Platten mit senkrechten, seitlichen Begrenzungen. Damit bleibt der einmal generierte Winkel der Photonspur zu Boden- und Deckelfläche erhalten. Für Winkel unter dem der Totalreflexion ($\sim 40^\circ$) nimmt

²⁶Durch die Verschiebung des Arguments wurde für den betrachteten Winkelbereich $[-90^\circ, 90^\circ]$, der Wertebereich des Cosinus auf das Intervall $[-1, 1]$ abgebildet. Der Ursprung ist für alle Richtungsvektoren in (0/0/0) gelegt.

²⁷Rot bedeutet maximal $40 \frac{\text{Photonen}}{\text{bin}}$ und blau minimal $1 \frac{\text{Photon}}{\text{bin}}$, während weiß leeren bins entspricht.

zum einen mit dem Winkel die Reflexionswahrscheinlichkeit schnell ab, zum anderen steigt die Anzahl der Reflexionen (siehe auch 2.3 (d))

- In y -Richtung besitzen die Photonen vorwiegend Winkel unter -30° , das heißt in der xy -Projektion bewegen sie sich, wie man bei der Form des 'Quirls' erwartet, im Mittel parallel zur Richtung der Tangente an die äußere Spirale.

(iii) Luminosität

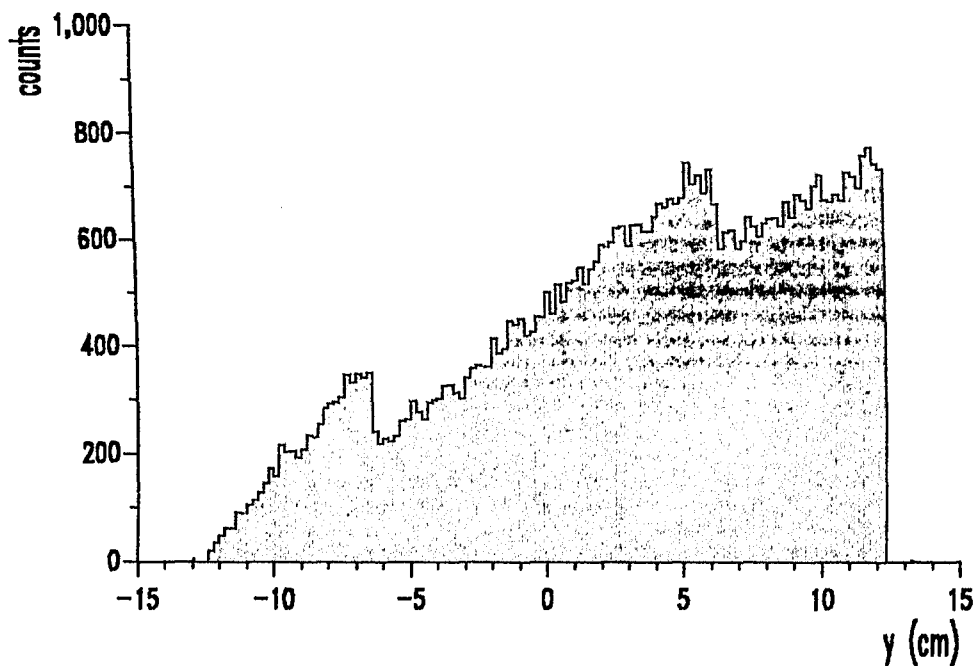


Abbildung 2.10: Luminosität der 'Quirl'- Stirnfläche.

Das Histogramm (Abbildung 2.10) gibt die Verteilung der ausgekoppelten Photonen über die Stirnfläche wieder²⁸. Zum äußeren Rand hin nimmt die Intensität der austretenden Photonen zu. Man erkennt weitere Stufen, die von der Form des angesetzten, trapezförmigen Quaders herrühren und zeigen, daß mit zunehmendem Winkel zwischen mittlerer Photonenrichtung (etwa zur Tangente an die äußere Spirale) und Auskopplungsfläche die Transmission besser wird²⁹ (siehe auch Figur 3.1.2).

²⁸Integriert über die z -Koordinate.

²⁹Für den Übergang Szintillator - Plexiglas ergibt das Verhältnis der Brechungsindizes einen Totalreflexionswinkel von etwa 70° . Die Photonen treffen im Mittel (s.o.) mit einem Winkel von

- (b) Den experimentellen Ergebnissen des nächsten Kapitels vorausgreifend, soll hier schon ein Vergleich dieser mit Simulationsergebnissen erfolgen.

(i) Zeitspektren

An den Punkten A3, D1 und E2 wurden jeweils 70 000 Photonen³⁰ lokal generiert. Zur Nomenklatur der lokalen Bestrahlungsorte siehe Kapitel 3.3 und Figur 3.1. Abbildung 2.11 zeigt die dazu berechneten Zeitspektren.

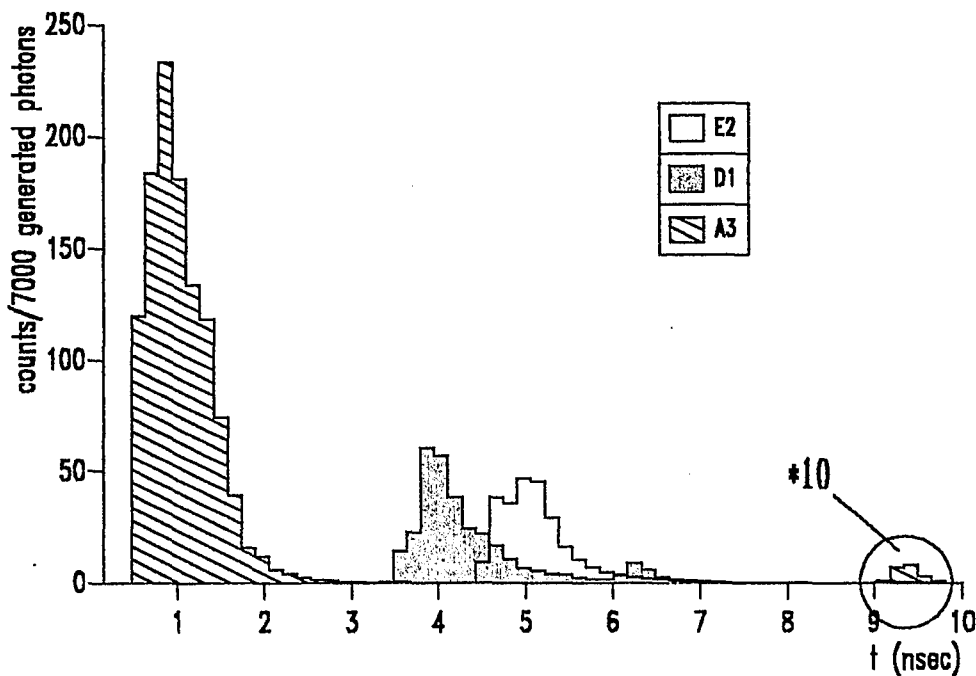


Abbildung 2.11: Zeitspektren, simuliert mit an den Punkten A3, D1 und E2 lokal generierten Photonen.

Das Zeitspektrum zu D1 (herausvergrößert auch das zu A3) zeigt ein zweites, niedrigeres Maximum. Es rührt von Photonen her, die sich zuerst in Richtung der 'Quirl'-Spitze bewegen und dann umgelenkt werden.

Vergleicht man die Positionen der Maxima mit dem Diagramm in Abbildung 3.26,

> 75° zum Lot auf die äußere Quaderfläche, die der Auskopplung dient! Der Winkel gegenüber den anderen beiden Flächen nimmt je um 15° ab.

³⁰Durchquert ein minimal ionisierendes Teilchen einen 3mm dicken Plastiksintillator, so entstehen etwa 7000 Photonen. Um eine bessere Statistik zu erhalten, wurden 70 000 Ereignisse generiert und die Ergebnisse um den Faktor 10 korrigiert. Die Anzahl der experimentell aufgenommenen Events betrug 100 000 (mit im Mittel 7000 erzeugten Photonen), was eine um 10⁴ bessere Statistik ergibt!

so ergibt sich innerhalb des Fehlers Übereinstimmung zwischen Experiment und Simulation³¹. In Bezug auf das spätere Experiment (mit einem Photomultiplier pro 'Quirl'-Segment) zeigt obige Abbildung die ungefähre Form, der zu erwartenden Photomultiplier-Ausgangssignale. Es muß dabei allerdings berücksichtigt werden, daß im Lichtleiter weitere Laufzeitverschiebungen für die Photonen auftreten werden. Auch sind dem wirklichen Signal noch die statistischen Prozesse, die im Photomultiplier ablaufen, überlagert.

(ii) Spezieller Versuchsaufbau

Um einen direkten Vergleich *Experiment* \leftrightarrow *Software* zu ermöglichen, wurde wie unter 2.2.4 beschrieben, der Versuchsaufbau genau simuliert. Die zu den 22 Meßpositionen jeweils berechneten 14 000 Photonenspuren stellen einen Kompromiß zwischen guter Statistik und aufgewendeter Rechenzeit dar.

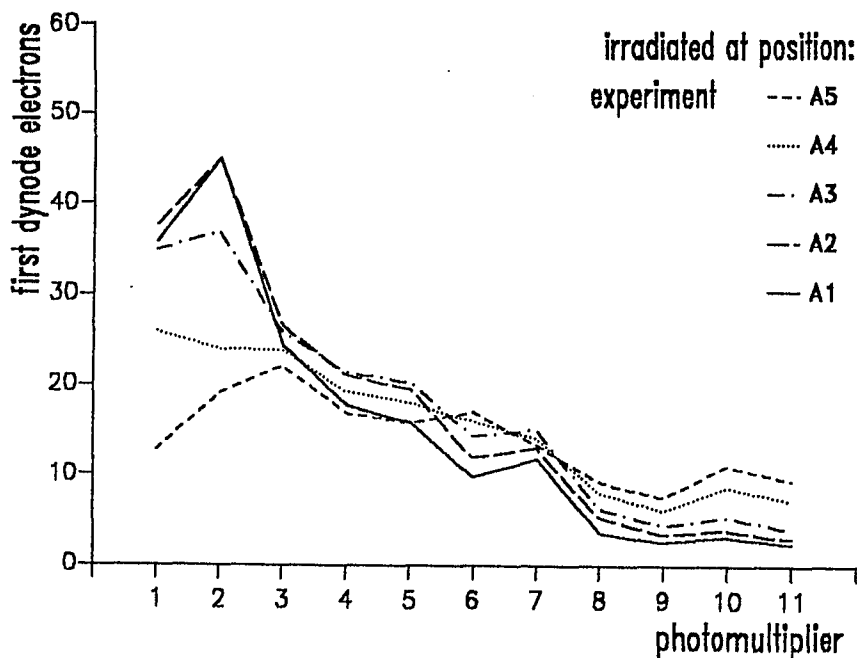


Abbildung 2.12: Simulierte Lichtausbeute der Photomultiplier aufgetragen gegen deren Position (normiert auf 7000 generierte Photonen).

³¹Die an der Stirnfläche reflektierten Photonen wurden hier nicht weiter verfolgt. Für D1 und E2 würde durch diese die Laufzeit im Mittel zu längeren Zeiten hin verschoben.

Diagramm 2.12 zeigt das Ergebnis³² für die Bestrahlungsorte A_i . Hierbei wurden berücksichtigt:

- (i) Die 'quantum efficiency' der Photomultiplier; nur jedes vierte auftreffende Photon wird im Mittel registriert.
- (ii) Oberflächenverluste; Photonen werden aufgrund verschmutzter oder verletzter Oberflächen nicht reflektiert. Diese Wahrscheinlichkeit (*surface loss*) betrug für die gezeigten Ergebnisse $\frac{1}{1000}$.
- (iii) Nicht ideale Klebeflächen; die Verluste wurden mit 15% angenommen³³.

Die beiden letzten Punkte (ii) und (iii) stellen die einzigen 'Fitparameter' des Programms dar und tragen dem realen Zustand des benutzten Szintillators Rechnung.

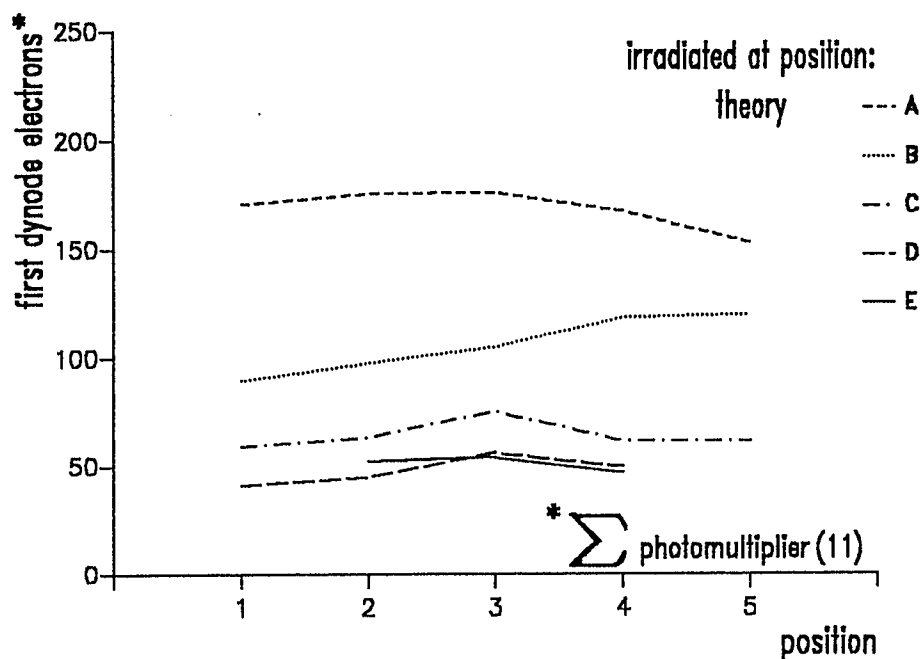


Abbildung 2.13: Simulierte integrale Lichtausbeute aufgetragen gegen die Bestrahlungsorte quer zum 'Quirl' (normiert auf 7000 generierte Photonen, siehe auch Abbildung 3.22).

³²Unter der Annahme, daß während des Experiments im Szintillator etwa 7000 Photonen pro Ereignis entstanden, wurden die Simulationsergebnisse mit dem Faktor 0.5 multipliziert.

³³Die Wahrscheinlichkeit, daß Photonen an den Flächen zwischen den Plexiglas-Ansatzstücken nicht reflektiert werden, wurde hier wie im Experiment qualitativ zu 50% abgeschätzt.

In Abbildung 2.13 ist die integrale Ausbeute (Summe über die 11 Photomultiplier) gegen die Photomultiplierposition quer zum 'Quirl' aufgetragen (analog zu Diagramm 3.22, Konfiguration 1).

Die gewählte Dämpfung durch Oberflächenverluste war mit $\frac{1}{1000}$ zu groß gewählt, so daß die Ausbeute für die hinteren Bestrahlungsorte (E_i , D_i und C_i) etwas zu niedrig ausfällt.

- Im Vergleich mit den experimentellen Daten ergibt sich qualitativ und quantitativ recht gute Übereinstimmung.

(c) Rechteckige Körper (Würfel, Stäbe ...)

Untersucht werden soll ein homogen beleuchteter, rechteckiger Szintillator (Brechungsindex 1.581 \Rightarrow Totalreflexionswinkel $\alpha_t = 39.24^\circ$). Es herrschen Idealbedingungen, nur Totalreflexion ist zugelassen und das Licht wird an einer Seite ausgekoppelt. Für diesen Spezialfall kann man die Wahrscheinlichkeit analytisch bestimmen, mit der ein Photon die Auskopplungsfläche erreicht³⁴.

In Figur 2.14 werden die Photonen im Punkt P erzeugt und zunächst nur eine (unendlich ausgedehnte) Fläche (E) betrachtet. Der eingezeichnete Kegel erfüllt die Bedingung:

$$\text{Öffnungswinkel} = \alpha_t,$$

daß heißt, Photonen deren Spur innerhalb des Kegels liegt, werden nicht (α), alle anderen (zum Beispiel β) werden an der Fläche reflektiert.

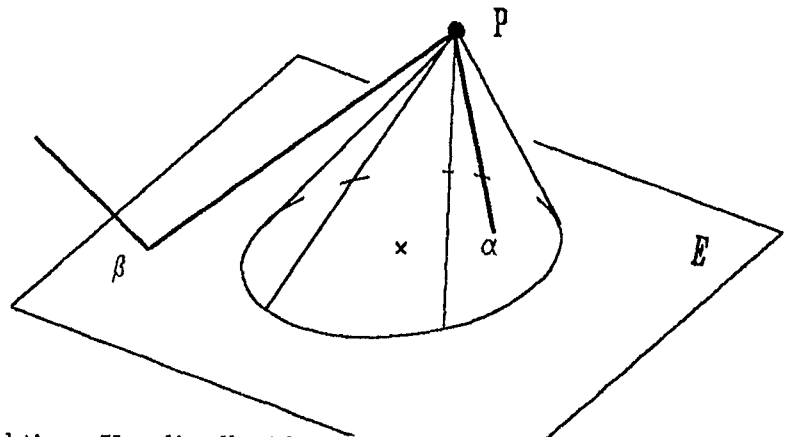


Abbildung 2.14: 'Totalreflektions-Kegel': alle Photonen, deren Spur innerhalb des gezeigten Kegels liegt, werden nicht an der Ebene E reflektiert.

³⁴Lediglich Photonen mit Winkel $> \alpha_t$ zum Lot auf eine Fläche werden an ihr reflektiert; es treten keine materialbedingten Verluste auf. Die Photonen sollen in einiger Entfernung von der Auskopplungsfläche erzeugt werden, so daß die später erklärten Seitenkegel diese nicht direkt treffen.

Der durch diesen Kegel aus dem gesamten Raumwinkel (4π) herausgeschnittene Anteil η beträgt:

$$\eta = \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\alpha_t} \sin \theta \, d\theta d\varphi = 11.27\%$$

Die Flächen des Körpers sind senkrecht zueinander vorausgesetzt. Da der Totalreflexionswinkel $\alpha_t < 45^\circ$ ist, können Photonen, die sich innerhalb des Kegels bewegen und auf eine Seitenfläche treffen (die Seitenfläche schneidet den Kegelmantel, siehe Figur 2.15 (γ)), an ihr nur reflektiert werden. Sie gelangen also sicher und mit unverändertem Winkel (im Vergleich zu oben, nur an der Seitenfläche gespiegelt) auf die Ebene E, an der sie nicht reflektiert werden.

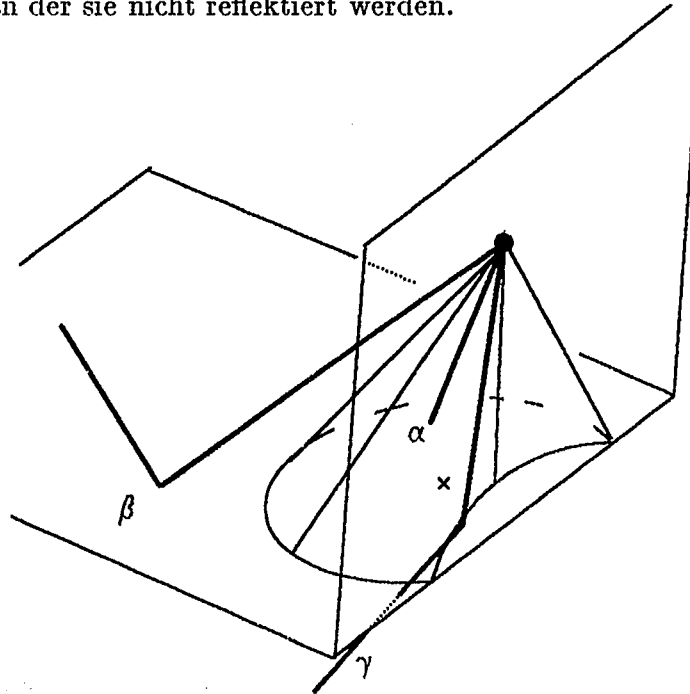


Abbildung 2.15: Durch Seitenfläche abgeschnittener 'Totalreflexionskegel'. Die Wahrscheinlichkeit für eine Reflexion an der Bodenfläche ist davon unabhängig.

Zu jeder der 6 Flächen des Körpers existiert ein solcher Kegel, unabhängig vom Ort der Photonerzeugung. Unter idealen Bedingungen würden damit $6 \cdot 11.27\%$ der Photonen den Szintillator verlassen, der Rest (32%) wäre in ihm gefangen.

Kontaktiert man nun eine Fläche³⁵, so bekommt man, allein durch Totalreflexion, 43.6% der erzeugten Photonen. Mit jeder weiteren kontaktierten Fläche erhöht sich der Prozentsatz der ausgekoppelten Lichtquanten um 11.27%.

³⁵Egal wie groß diese ist, z.B. auch die kleine Stirnseite eines dünnen, langen Stabes!

Mit dem Programm wurden verschiedene rechteckige Körper simuliert:

- Von 100 000 generierten Photonen erreichten, in guter Übereinstimmung mit obigen theoretischen Betrachtungen, rund 43 200 die Auskopplungsfläche.

Die berechnete Winkelverteilung, für die Photonen an der Stirnfläche eines Stabes ist in Abbildung 2.16 dargestellt.

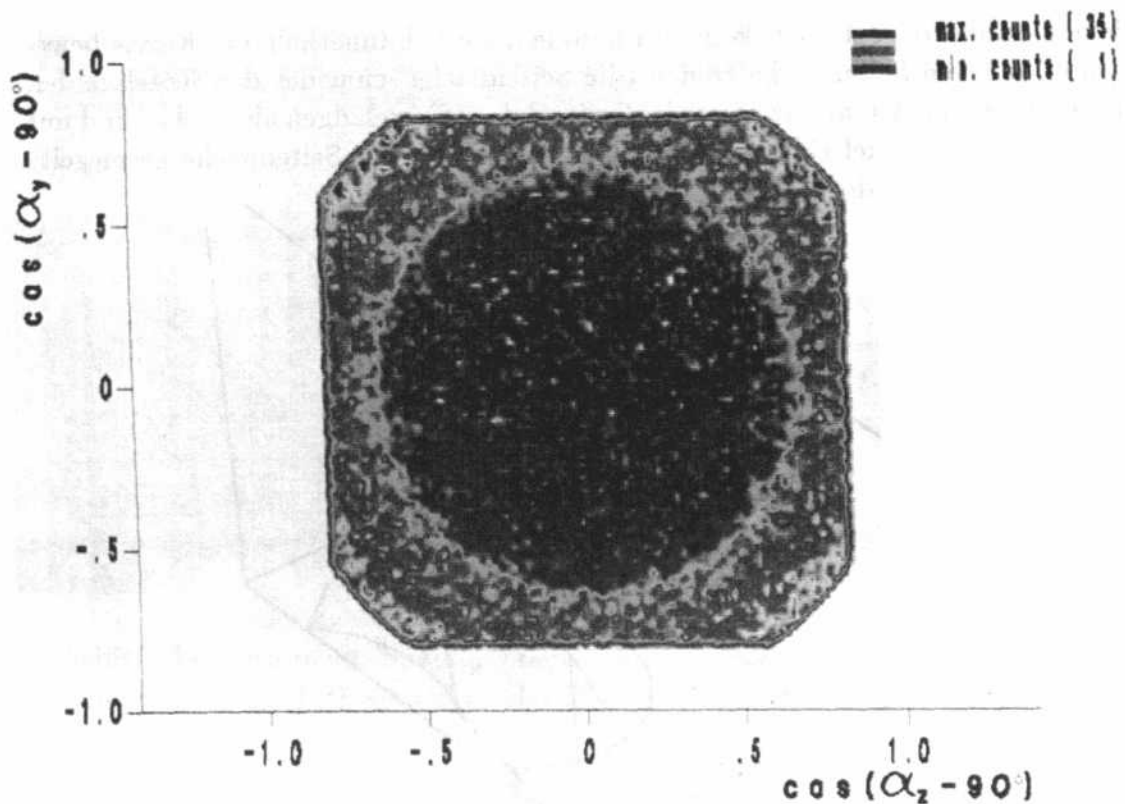


Abbildung 2.16: Winkelverteilung der Photonen an der Auskopplungsfläche eines rechteckigen Stabes. Dargestellt analog zur Winkelverteilung des 'Quirls'.

Dieses Diagramm ermöglicht einen Vergleich mit der Winkelverteilung der Photonen an der 'Quirl'-Stirnfläche (Abbildung 2.9)³⁶. Man erkennt, daß die Photonen den Stab divergenter verlassen; viele liegen mit ihrer Richtung zwischen zwei Kegeln, mit den Öffnungswinkeln von 39° bis 45°. ³⁷

³⁶Um diesen Vergleich zu ermöglichen, wurde hier bei der Simulation im Stab, die Reflexion wieder mit Hilfe der Fresnel-Formeln berechnet.

³⁷Der innere, dunkelblaue Kreis rührt von den 11.27% Photonen her, deren Spur innerhalb des

(d) Ergänzende Ergebnisse zum 'Quirl'

In diesem Abschnitt soll die berechnete Lichtausbeute des bisher betrachteten 'Quirls', mit der dreier anderer Szintillatorformen verglichen werden. Die Photonen (70 000) wurden homogen in dem jeweiligen Körper erzeugt.

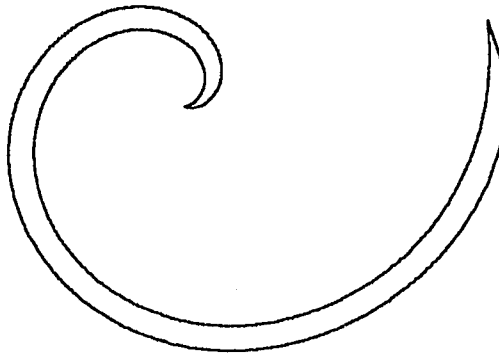
- (i) Statt der bisherigen Dicke von $3mm$, ist der ansonst gleiche 'Quirl' nun $5mm$ dick. Es werden jedoch für den Vergleich ('*unphysikalisch*') gleichviele Photonen generiert.

$$\nu = \frac{Ausbeute'Quirl'_{3mm}}{Ausbeute'Quirl'_{5mm}} = 0.93$$

Die Weglänge im Szintillator ändert sich kaum, nur die Anzahl der Reflexionen an Boden- und Deckelfläche nehmen um etwa $\frac{2}{5}$ ab.

- (ii) Der maximale Winkel φ beträgt statt 180° jetzt 360° (siehe Abbildung).

$$\nu = \frac{Ausbeute'Quirl'_{180^\circ}}{Ausbeute'Quirl'_{360^\circ}} = 1.79$$



Hier mußte die Routine *Newton* umgeschrieben werden, da jetzt mehr als 2 Schnittpunkte einer Geraden mit einer der Spiralen möglich sind.

'Totalreflexionskegels' für die Stirnfläche liegt. Die Photonen, die innerhalb der zu den Seitenflächen (diese stehen zur Auskopplungsfläche senkrecht) gehörenden 'Totalreflexionskegel' liegen und damit 'verloren gehen', bedingen die fehlenden vier Kreissegmente (Kreisabschnitte).

- (iii) Der dritte Szintillator hatte die Form eines Tortenstücks. Die Maße waren denen des 'Quirl' angepaßt, die Dicke betrug $3mm$.

$$\nu = \frac{Ausbeute'Quirl'}{AusbeuteTortenstück} = 0.64$$

Der 'Quirl' erweist sich mit $2/3$ der Ausbeute des Tortenstücks als **guter Lichtleiter**³⁸.

³⁸Die Rechnung ergibt weiter, daß die Stirnfläche des Tortenstücks völlig homogen ausgeleuchtet wird!

Kapitel 3

Experiment

Ziel ist es, experimentelle Aussagen über die optischen Eigenschaften eines 'Quirl'-Segments zu erlangen, um die Form des Detektors, speziell im Hinblick auf seine Lichtausbeute zu optimieren. Für die Dicke des Segments ist ein Kompromiß gesucht: Sie sollte hinlänglich sein, um Signale zu erhalten, die eine gute Zeitauflösung garantieren. Andererseits möchte man, um die kinematischen Parameter der Reaktionsteilchen nur gering zu beeinflussen, dem Ideal des 'masselosen Detektors' möglichst nahe kommen.

Auf die Meßergebnisse wird am Ende des Kapitels eingegangen. Ein Vergleich mit den Simulationsrechnungen folgt in Kapitel 4.

3.1 Versuchsaufbau

Die Maße des in dieser Arbeit untersuchten 'Quirl'-segments (es ist ein Teil der JETSET-Trigger Endkappe) betragen:

- maximaler Radius: 48cm , • Dicke: 3mm ,
- Öffnungswinkel: 30° , • maximaler Winkel φ : 180°

und die Randkurven gehorchen, wie schon unter 2.2 dargelegt den Gleichungen

$$x = \frac{r_{\max}\varphi}{\varphi_{\max}}\cos(\varphi + \Delta) , \quad y = \frac{r_{\max}\varphi}{\varphi_{\max}}\sin(\varphi + \Delta)$$

mit $\varphi \in [0^\circ, 180^\circ]$, $\varphi_{\max} = 180^\circ$, $\Delta = 0^\circ$ und $\Delta = 30^\circ$.

Der verwendete Szintillator wurde in der mechanischen Werkstatt des IKP¹ bearbeitet.

¹Institut für Kernphysik der Kernforschungsanlage Jülich, NC-Fräßbank der Firma Maho MH600C

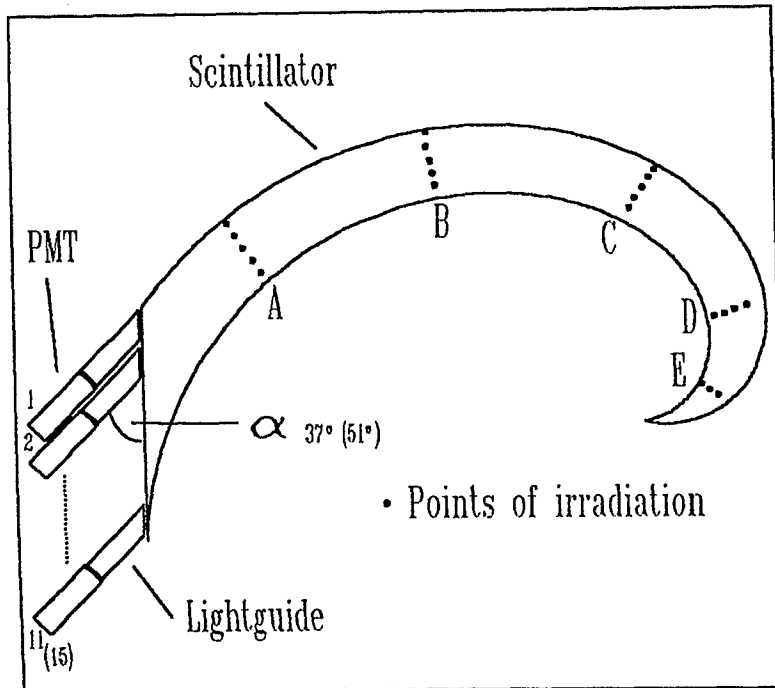


Abbildung 3.1: Schema des Versuchsaufbau mit vereinfacht dargestellter Stirnfläche

Die Stirnfläche wurde mit Photomultipliern bestückt, die über parallele Plexiglas²-Lichtleiter angekoppelt waren. So erhält man neben der integralen Information, wie Lichtausbeute in Abhängigkeit des Bestrahlungsortes, auch die Verteilung der Photonen über die Flächen, welche später der Lichtauskopplung dienen sollen. Um zusätzlich die erwartete Richtungsverteilung der Photonen überprüfen zu können, wurden zwei Winkel α durchgemessen³. Der gesamte Aufbau lagerte auf 5 cm hohen Sockeln innerhalb einer schwarz ausgekleideten, lichtdichten Aluminiumkiste. Dies ermöglichte die in einem Kollimator befindliche β -Quelle und die beiden Szintillationszähler des Triggers einfach auf die in Abbildung 3.1 skizzierten Punkte zu positionieren.

3.1.1 Szintillator

Benutzt wurde das Plastiksintillatormaterial BC 404 der Firma BICRON, dessen physikalisch relevanten Daten in Tabelle 3.1 zusammengestellt sind. Da mit Hilfe des Szintillators später der zeitliche Startpunkt für die anderen Komponenten des kompakten Detektorsystems JETSET bestimmt werden soll, wurde bei seiner Wahl

²reg. Warenzeichen der Röhm GmbH, Darmstadt

³ $\alpha_1 = 37^\circ$ entspricht etwa dem erwarteten günstigsten Wert; $\alpha_2 = 51^\circ$ wurde als Vergleichspunkt gewählt.

Light Output % Anthracene	68
Refractive Index	1.581
Rise Time <i>ns</i>	0.7
Decay Time <i>ns</i>	1.8
pulse width (Full Width Half Maximum) <i>ns</i>	2.2
Light attenuation Length <i>cm</i>	120
Wavelength of Max. Emission <i>nm</i>	408
No. of C Atoms per cm^3 ($\cdot 10^{22}$)	4.74
No. of H Atoms per cm^3 ($\cdot 10^{22}$)	5.21
Ratio H:C Atoms	1.100
No. of Electrons per cm^3	3.37
Principal Applications	fast counting

Tabelle 3.1: Materialkonstanten für BC 404 [BIC86]

besonderer Wert auf die kurze Signalbreite in Verbindung mit der Anstiegszeit und eine hohe Lichtausbeute⁴ gelegt. Zur Auswahl standen 3 Szintillatormaterialien mit ähnlich guten Eigenschaften. Die Vergleichsmessungen, die zur Wahl des BC 404 führten, sind in [Sef88] beschrieben.

Aromatische Kohlenwasserstoffverbindungen, zum Beispiel Benzol, bilden zwischen den C-Atomen neben den σ -Bindungen noch π -Bindungen aus. Die Anregungszustände solcher delokalisierten π -Elektronensysteme sind für den Lumineszenzeffekt verantwortlich. Als Lumineszenz wird die Emission von Licht mit einem charakteristischen Spektrum nach der Absorption normalerweise höherenergetischer Strahlung bezeichnet.

Plastikszintillatoren sind Zusammensetzungen solcher aromatischer Verbindungen. In einem Polymer, das als Trägerstoff dient und selbst schon szintilliert - bei dem benutzten BC 404 ist dies Polyvinyltoluol (PVT) - sind geringe Mengen (weniger als 4% Gewichtsanteil) besonderer Aromate gelöst, die auch Wellenlängenschieber genannt werden. Zwei solche für Szintillatoren verwendete Moleküle sind PPO⁵ und POPOP⁶. Sie vergrößern die absolute Szintillationseffizienz S , den nicht als Wärme sondern als Licht abgegebenen Anteil der deponierten Energie, erheblich, so daß S in

⁴Je mehr Photonen den Photomultiplier erreichen, desto besser wird seine Zeitauflösung, da sich der Time-jitter der Elektronen mittelt. Für die Auflösung t ergibt sich etwa: $t \sim \frac{1}{\sqrt{n}}$ [LC77].

Analog gilt dies auch für die Photonen und deren Laufzeit im 'Quirl'.

⁵2,5 -phenyloxazol

⁶1,4-di-[2-(5-phenyloxazolyl)]-benzol

nach der neuen IUPAC-Nomenklatur: Benzen,Toluen

der Größenordnung von einigen Prozent liegt⁷. Erreicht wird dies durch Verschieben der Absorptions- und Emissionsspektren gegeneinander, wodurch der Szintillator für das von ihm erzeugte Licht durchsichtiger wird. Mit geeigneter Wahl der Zusätze kann das Emissionsspektrum so auch an die spektrale Empfindlichkeit der Photomultiplier angepaßt werden. Abbildung 3.2 zeigt das Emissionsspektrum von BC 404.

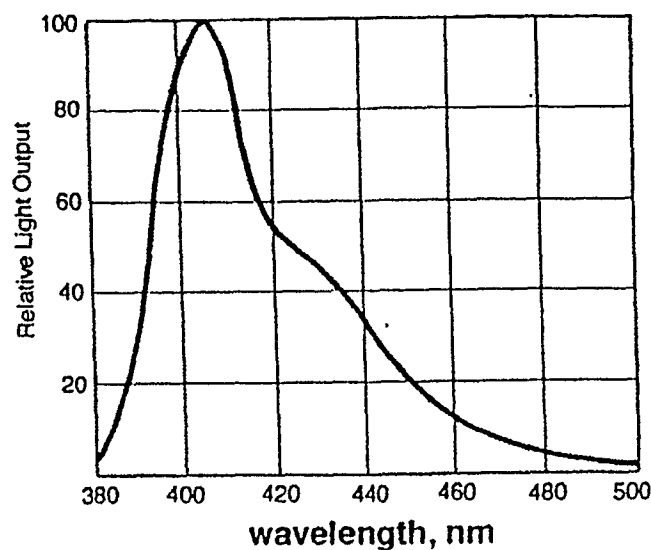


Abbildung 3.2: BC 404 Emissionsspektrum [BIC74]

Der Szintillationsmechanismus kann in zwei unabhängige Prozesse aufgeteilt werden:

- (i) Durch die ionisierende Strahlung wird Energie im Szintillator deponiert, welche fast ausschließlich in Form von Anregungsenergie des Polymers vorliegt, da die Konzentration der Beimischungen sehr gering ist.
- (ii) Es finden konkurrierende Prozesse um diese Anregungsenergie statt. Übertrag der Energie auf andere Moleküle des Polymers oder der zugegebenen Aromate, sowohl thermisch als auch über Strahlung - innere Konversion in Wärme - Quenching⁸ und Emission als Fluoreszenzlicht.

Zur Pulsbreite (FWHM) tragen verbreiternd neben der Fluoreszenz auch noch Phosphoreszenz und verzögerte Fluoreszenz bei. [Bir53] [Hur85] [Tho83]

⁷Die Lichtausbeute liegt für Plastiksintillatoren bei etwa $\frac{1 \text{ Photon}}{100 \text{ eV deponierte Energie}}$.

⁸Quenching kann als Sättigungseffekt mit verschiedenen Ursachen beschrieben werden, wobei deponierte Energie ebenfalls statt in Licht in Wärme umgewandelt wird. Eine ausführliche Beschreibung findet man in [Bir67]

3.1.2 Lichtleiter (Montage)

Betrachtet man die Geometrie⁹ des Segments, so erwartet man in der Projektion im Mittel Richtungen der austretenden Photonen, die zwischen den Tangenten an die Spiralen liegen. Dies entspricht Winkeln zur Verbindungsgeraden V der Spiralenendpunkte von $\alpha \leq 30^\circ$ (Abbildung 3.3 (a)). Die beiden angefertigten Konfigurationen mit:

$$\alpha_1 = 37^\circ, \alpha_2 = 51^\circ$$

stellen einen Kompromiß dar. Diese Anordnung erlaubte die 3 Flächen $F1, F2, F3$ gleichmäßig mit Photomultipliern zu besetzen.

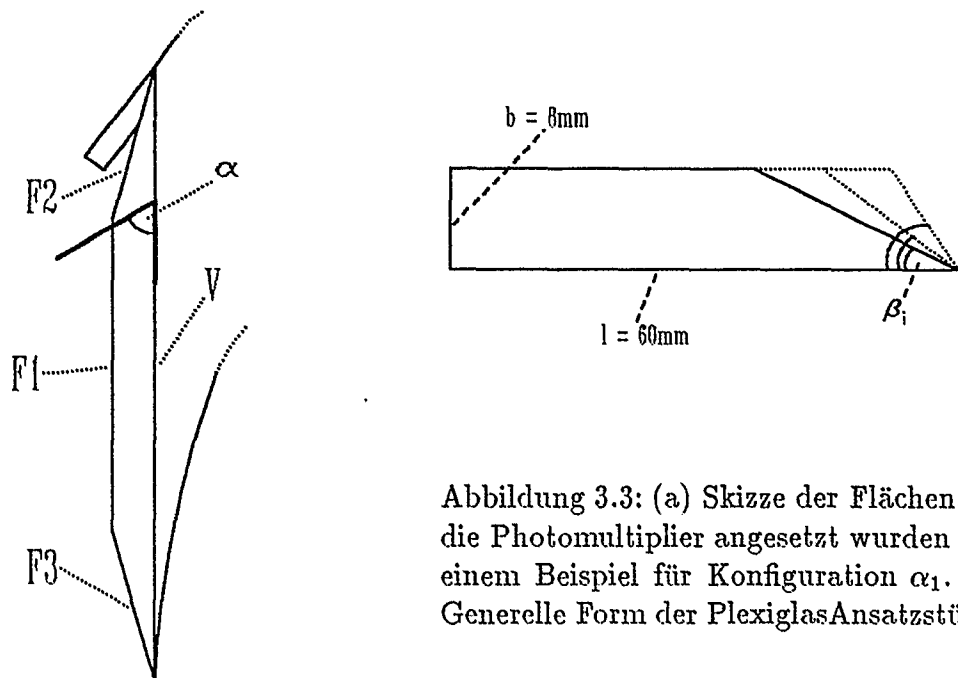


Abbildung 3.3: (a) Skizze der Flächen, an die Photomultiplier angesetzt wurden mit einem Beispiel für Konfiguration α_1 . (b) Generelle Form der Plexiglas-Ansatzstücke

Als Verbindung zwischen Szintillator und Photomultipliern wurden Ansatzstücke aus Plexiglas hergestellt (PMMA¹⁰, Brechungsindex 1.49). Dicke 3mm , Breite festgelegt durch den Kathodendurchmesser des benutzten Photomultipliertyps zu 8mm , Länge der Seite l konstant 60mm . Der Fehler im Winkel β konnte trotz manueller Fertigung unter $\pm 50^\circ$ gehalten werden.

Zur Vermeidung einer komplizierten Halte- und Anpreßvorrichtung fanden Unter-

⁹zur Geometrie des 'Quirls' siehe Kapitel 2.2

¹⁰Plexiglas GS DIN 4102-B2, Polymethylmethacrylat

suchungen statt, mit dem Ziel optische Kontakte durch reversible Klebungen mit genügend mechanischem Halt zu erzeugen. Zwei Methoden lieferten zufriedenstellende Ergebnisse:

- Für größere Querschnittsflächen reichte ein 'Silikonkautschuk'¹¹ aus, der etwa 4 Stunden (bei Zimmertemperatur) zum Aushärten benötigt und allein durch mechanische Einwirkung wieder rückstandslos zu entfernen ist. Dieser wurde bei der Konstruktion der Triggeranordnung verwendet.
- Für die kleinen Klebeflächen der Ansatzstücke war dagegen ein optischer Zement der Firma Nuclear Enterprises¹² zu bevorzugen, mit einer Aushärtezeit von 24 Stunden (ebenfalls bei Zimmertemperatur). Er konnte erst nach mehreren Tests¹³ eingesetzt werden und erforderte eine sorgfältige Reinigung der gebrochenen Klebestellen. Zum Anlösen der Zementreste eignet sich am besten Methanol¹⁴.

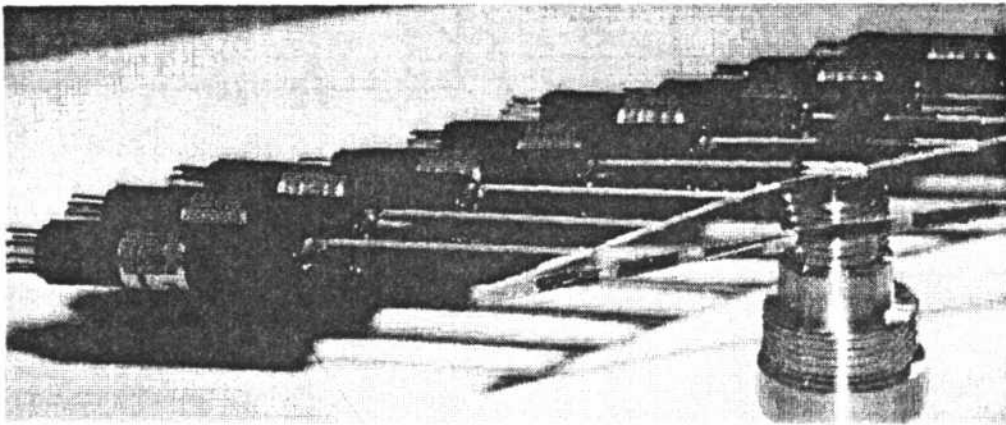


Abbildung 3.4: Fertiger Aufbau der 2. Konfiguration

Trotzdem zeigte die Erfahrung, daß dies eine recht zuverlässige Methode ist, die komplizierte Aufbauten vermeiden hilft (immerhin mußten 11 beziehungsweise 15 Photomultiplier befestigt werden).

¹¹Wacker Silicone, Silgel 604A

¹²Nuclear Enterprises Limited, optical cement NE581

¹³Diese wurden an Plexiglasmodellen, einem defekten und erst nach einem Langzeittest auch an einem intakten Photomultiplier, bis zu einer Klebefläche von 60mm^2 durchgeführt.

¹⁴Methanol wird durch die Haut vom Körper resorbiert! Ersatzweise kann auch Ethanol verwendet werden.

Um gleichmäßige Klebungen zu erhalten, wurden die Plexiglasansatzstücke zum Kleben so gelagert, daß sie durch ihr Eigengewicht plan angedrückt wurden. In einem zweiten Arbeitsgang standen die Photomultiplier zum Kleben frei auf den Endflächen der Verbindungsstücke.

Aus fertigungstechnischen Gründen betrug der Abstand zwischen den Lichtleitern etwa 3mm ($\pm 1\text{mm}$). Sie waren während der Messung durch Aluminiumfolie optisch voneinander getrennt.

3.1.3 Photomultiplier

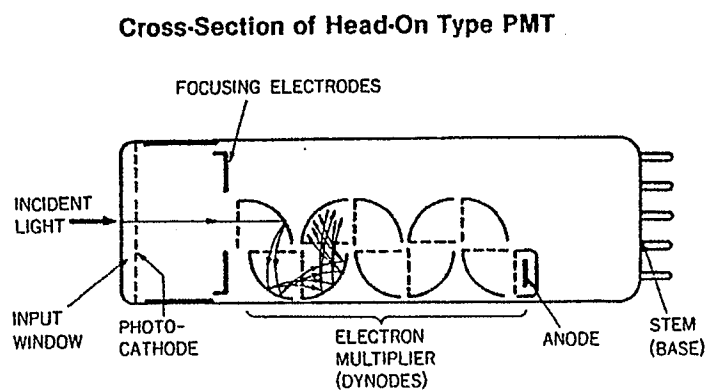


Abbildung 3.5: Schema einer Photomultierröhre [HAM88]

Abbildung 3.5 zeigt den prinzipiellen Aufbau einer Photomultierröhre. Sie setzt sich aus zwei Hauptkomponenten zusammen,

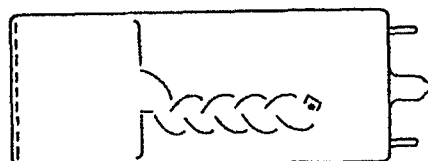
- eine lichtempfindliche Photokathode
- und ein Elektronenvervielfacher

die sich in einem evakuierten Glaskolben befinden. Fällt Licht auf die Photokathode, so emittiert diese mit einer gewissen Wahrscheinlichkeit (\rightarrow quantum efficiency) Elektronen, welche durch Fokussierelektroden auf die erste Dynode gelenkt werden. Dort und auf jeder weiteren werden pro auftreffendes Elektron mehrere Elektronen durch Sekundäremission freigesetzt, die aufgrund einer anliegenden Potentialdifferenz zur folgenden Dynodenstufe hin weiterbeschleunigt werden. Das Ende der Kette bildet die Anode, welche die vervielfachten Elektronen als Ausgangssignal sammelt.

Für das Experiment wurden 'linear focused, head on' Photomultiplerröhren der Firma HAMAMATSU¹⁵(R1635-02), wie rechts skizziert, verwendet. Die Maße der Röhren betragen:

- Länge 45mm
- aktiver Kathodendurchmesser 8mm
- Durchmesser 10mm

Linear Focused Type



Das Datenblatt mit für diese Röhren typischen Werten ist in Tabelle 3.2 wiedergegeben. Anhang A enthält Beispiele der zu jedem Photomultiplier mitgelieferten individuellen Informationen und Angaben über die zugehörigen Basen.

Type No.	R1635-02
Spectral Response	
Range nm	300 - 650
Peak Wavelength nm	420
Maximum Ratings	
Anode to Cathode Voltage V	1500
Average Anode Current mA	0.03
Cathode Sensitivity Luminous	
Min. $\mu A/lm$	60
Typ. $\mu A/lm$	90
Anode to Cathode Supply Voltage V	1250
Anode Sensitivity Luminous	
Min. A/lm	50
Typ. A/lm	200
Current Amplification Typ.	$2.2 * 10^6$
Anode Dark Current (after 30 min.)	
Typ. cps	100
Max. cps	400
Time Response	
Rise Time Typ. ns	0.8
Electron Transit Time Typ. ns	7.8

Tabelle 3.2: Typische Daten der benutzten Photomultiplier

Die spektrale Empfindlichkeit (spectral respons) der Kathode wird im kurzwelligen Bereich durch das Fenstermaterial **Borosilikatglas** und im langwelligen durch

¹⁵HAMAMATSU PHOTONICS K.K. Toyooka Village, Iwata-Gun, Shizuoka Pref., Japan

das Kathodenmaterial **Bialkali** begrenzt. In Abbildung 3.6 ist eine typische Verteilung für die benutzten Photomultiplier dargestellt. Bei guten Kathoden liegt die Quantenausbeute (quantum efficiency) heute bei etwa 28%. Der Vergleich mit dem Emissionsspektrum des Szintillators (→ Abbildung 3.2) zeigt dessen gute Anpassung.

Typical Spectral Response of Bialkali Photocathode (R878, etc.)

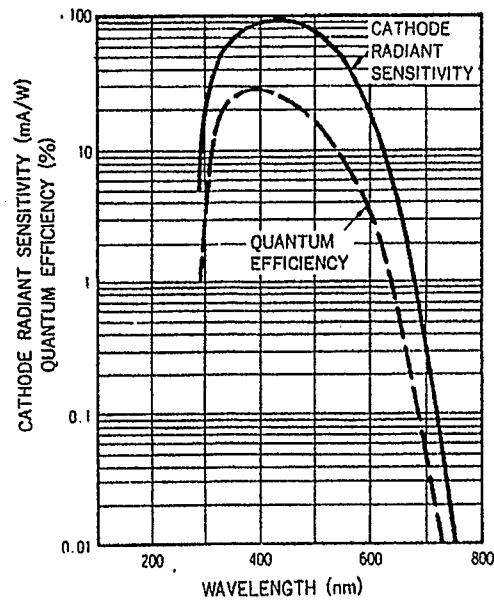


Abbildung 3.6: Spektrale Empfindlichkeit einer Bialkali Photokathode - typisch für die verwendeten Photomultiplier¹⁵. [HAM88]

Die R1635-02 besitzen eine Verstärkung¹⁶ von ungefähr $2.2 \cdot 10^6$. Da die Verstärkung schon von kleinen Magnetfeldern stark beeinflusst wird, ist der Glaskolben mit einem Magnetschild (μ -Metall) umgeben. Rise Time und Transit Time sind wegen der kleinen Ausmaße der Röhren, der wenigen Dynodenstufen (8) und der linearen Fokussierung sehr kurz. Der Dunkelstrom¹⁷ ist extrem niedrig (siehe Anhang A), so

¹⁵Die quantum efficiency ist als Verhältniss von emittierten Photoelektronen zu einfallenden Photonen definiert. Der Zusammenhang mit der Strahlungsempfindlichkeit (radiant sensitivity) lautet: $QE = \frac{S \cdot 1240}{\lambda} \cdot 100\%$

¹⁶Dieser Wert gilt für die empfohlene Versorgungsspannung von $-1250V$, die auch verwendet wurde.

¹⁷Ursachen für diesen Strom, der auch dann vorhanden ist wenn die Röhren in absoluter Dunkelheit betrieben werden, sind neben der thermischen Emission von Elektronen aus der Kathode

daß diese Photomultiplier zum Nachweis einzelner Photonen (single photon counting) eingesetzt werden können.

Zum Schluß dieses Abschnitts ist noch ein Bild dreier HAMAMATSU R1635-02 Photomultiplier zusammen mit einem 10 Pfennigstück wiedergegeben, um ihre Größe anzudeuten.

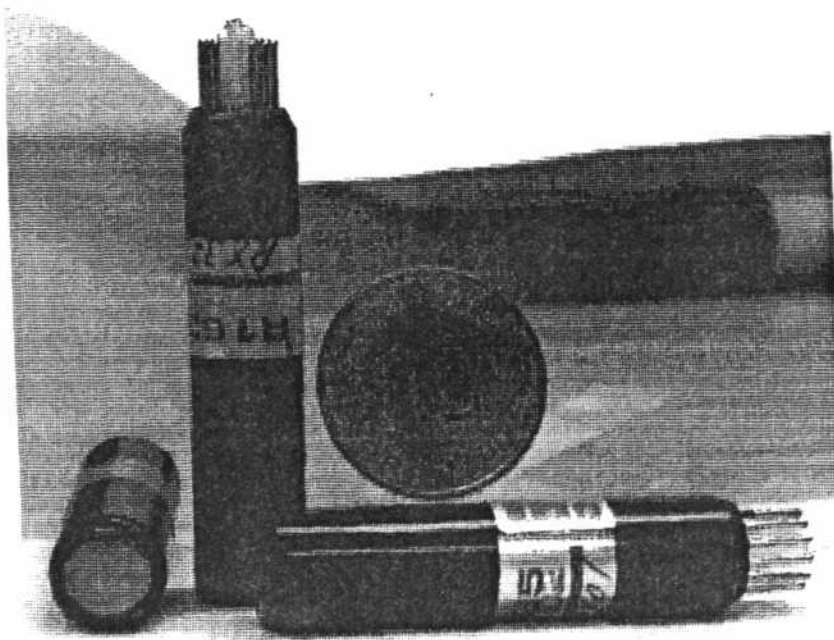


Abbildung 3.7: Photomultiplier HAMAMATSU R1635-02

[HAM88b] [VAL70] [RCA80]

3.1.4 Elektronik

Den Kern des Datenaufnahmesystems bildeten ein ADC- und ein TDC-Modul. Sie wurden via CAMAC-Bus von einem PC aus gesteuert und ausgelesen. Die dazu erstellte Software, Angaben zum Interface CAMAC/PC (DSP) und einige Hinweise

noch: (a) Ionisation der noch im Glaskolben vorhandenen Restgase. (b) Glasszintillation, wenn Elektronen von ihrem Weg abkommen (um dem entgegen zu wirken ist der Glaskolben mit einer leitenden Farbe überzogen, die auf Kathodenpotential liegt - HA Coating). (c) Ohmsche Verluste, die durch ungenügende Isolation im Bereich des Sockels auftreten. (d) Feldemission an Elektroden (nur wenn die Röhre im Grenzbereich betrieben wird).

zur Bedienung sind in Anhang B zusammengestellt.

ADC	CAMAC Modul 4300B 16 Channel, Fast Encoding and Readout Adc (FERA)
Fast Conversion:	8.5 μsec (11 bit)
Input Sensing:	Charge (current integrating)
High Sensitivity:	480 pC (11 bit), 0.25 pC/Channel
Input Impedance:	50 Ω , NIM level
Gate:	width 50-500 nsec, one for all ADCs, must precede input signals by at least 20 nsec
TDC	CAMAC Modul 2228A 8 Channel, Time to Digital Converter (TDC)
Dynamic:	11 bits
Conversion Time:	100 μsec (11 bit)
Input:	50 Ω , NIM level
Time Resolution:	50/100/250 psec
Time Range:	0..100/0..200/0..500 nsec

Tabelle 3.3: Spezifikationen der CAMAC Module, LeCroy4300B [LEC85] & LeCroy2228A [LEC86]

Das Gate für den FERA und das Startsignal für den TDC, der im 'common start/individual stop' Mode betrieben wurde, erzeugte in Verbindung mit dem Trigger (→ Kapitel 3.2) folgende Schaltung:

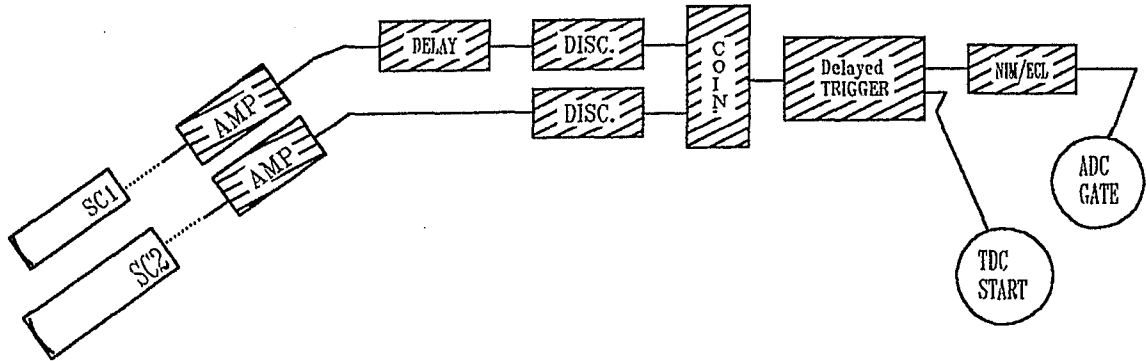


Abbildung 3.8: Blockschaltbild der Elektronik im Triggerzweig

Hierbei war das DELAY im SC1-Zweig nötig, um Laufzeitdifferenzen in den beiden Szintillationszählern (Länge der Lichtleiter, 'transit time' der Photomultiplier) auszugleichen. Mit Hilfe der Diskriminatoren erhielt man aus den verstärkten Röhrensignalen Normpulse mit einer Signalbreite (width) von etwa 5 nsec , die dann auf eine Koinzidenzeinheit gegeben wurden. Der folgende 'delayed Trigger' ermöglichte die Breite des Gates einzustellen und dieses zu verzögern. Die vordere, fallende Flanke konnte direkt als Startsignal für den TDC benutzt werden, während das Gate für den ADC ('command bus' in ECL Norm) noch durch einen NIM/ECL-Konverter angepaßt werden mußte.

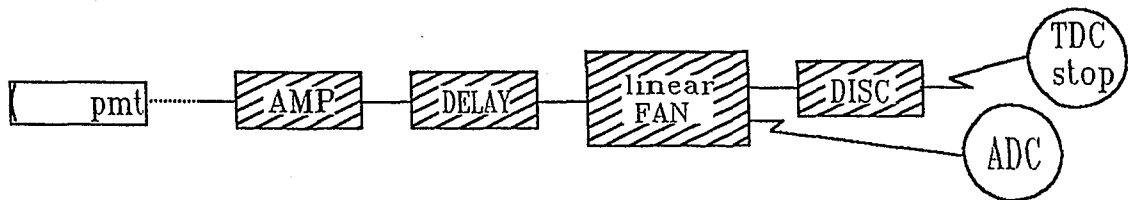


Abbildung 3.9: Blockschaltbild der Elektronik im Signalzweig

Abbildung 3.9 zeigt die Elektronik des Signalzweiges schematisch. Um die gesamte Dynamik der ADCs zu nutzen, mußten die Signale der Photomultiplier verstärkt werden (schnelle Photomultiplier Vorverstärker). Koinzidenz und Pulsformung im Triggerzweig erforderten etwa 45 nsec , so daß mit dem für das Gate geforderten Zeitabstand (siehe Tabelle 3.3) ein DELAY von grob 70 nsec folgte (Lemo-Kabel). Der zeitliche Feinabgleich geschah über den 'delayed Trigger'. Anschließend wurde das Signal in einem 'linear FAN' aufgespaltet, wodurch es auf den ADC gegeben werden konnte und gleichzeitig über einen Diskriminator das individuelle STOP für einen TDC-Kanal erzeugte ¹⁸.

Bis auf den 'delayed Trigger' von der Firma Struck waren alle ('fast'-)NIM Module von LeCroy. Daten zu diesen finden sich auch in Anhang B.

3.2 Trigger

Im folgenden wird der gesamte Aufbau, der es ermöglichte im 'Quirl' lokal Photonen zu erzeugen, als Trigger bezeichnet werden.

¹⁸Der beschriebene Aufbau existierte achtfach parallel. Die Photomultiplier für die kein TDC-Kanal vorhanden war, wurden über AMP und DELAY direkt mit dem Eingang des ADC verbunden.

3.2.1 Aufbau

Der Trigger bestand aus 3 Komponenten, einem Kollimator mit Quelle und zwei Szintillationszählern (*SC1*, *SC2*). In Abbildung 3.10 ist sowohl ein Foto als auch eine Skizze der gesamten Anordnung wiedergegeben.

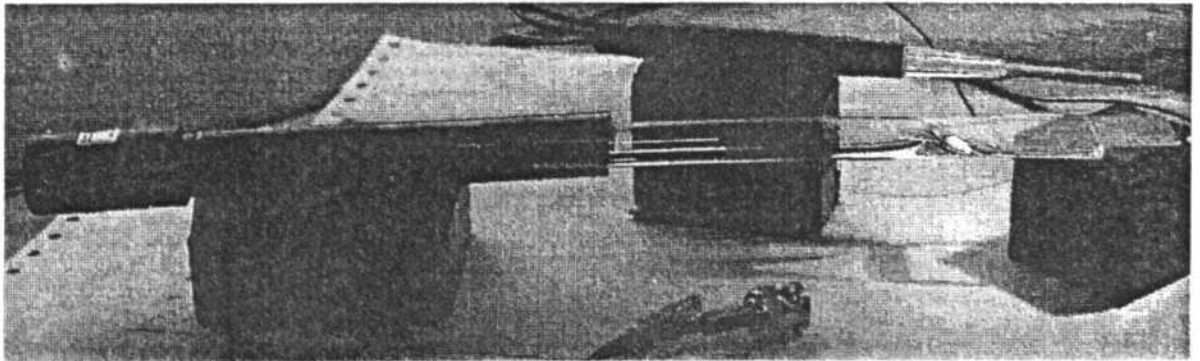
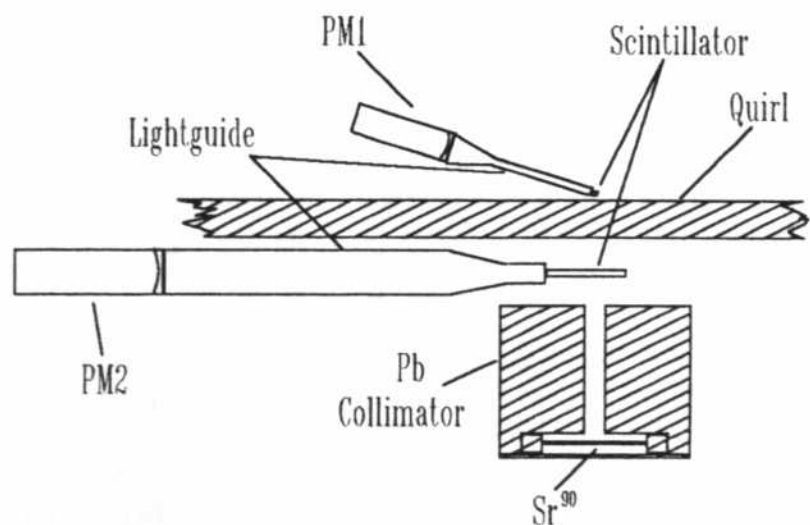


Abbildung 3.10: Foto und schematische Zeichnung der gesamten Triggeranordnung

Als Quelle stand ein Strontium Präparat mit einer Aktivität von $100 \mu\text{Ci}$ ¹⁹ zur Verfügung. Die Rate der in den vom Kollimator erlaubten Raumwinkel emittierten Elektronen betrug rund 10^4 pro Sekunde ²⁰. Es handelt sich bei Strontium um

¹⁹dem entsprechen $3.7 \cdot 10^6$ Zerfälle/sec

²⁰Bei der Berechnung der Rate wurden gestreute Elektronen nicht berücksichtigt.

einen reinen β -Stahler: [Gro88]

Nuclei	$t_{1/2}$ (y)	decay mode	E(KeV) / intensity(%)
^{90}Sr	28.5	$\beta^- \rightarrow \beta^-$	587 / 100
(^{90}Y)			2287 / 100

Zur Messung wurde der Kollimator mit dem großflächigen Szintillator von SC2 unter dem 'Quirl' positioniert, wie in Abbildung 3.10 gezeigt. Die eigentliche Ortsbestimmung sollte der kleine 'Szintillatorwürfel' von SC1 leisten. Nur wenn Elektronen in SC1 und SC2 koinzident ein Signal erzeugten, also auch wirklich den 'Quirl' durchquert hatten, sollte das GATE gesetzt²¹ werden. Mit dieser Forderung, die neben der geometrischen auch eine energetische Selektion der Elektronen bedingte, wurde eine Datenaufnahmerate von etwa 30 Ereignissen pro Sekunde gemessen.

Die geometrischen Daten des Triggers waren:

Kollimator:	Material <i>Blei</i> , Bodenfläche $45 * 45\text{mm}^2$, Höhe 38mm , Durchmesser der Bohrung 3.4mm , Tiefe der Bohrung 35mm .
SC1:	Photomultiplier (PM1) R1635-02 <i>Hamamatsu</i> , Szintillator NE102a ($1.8 * 1.8 * 1.3\text{mm}^3$), Abstand Kathode/Szintillator 7cm .
SC2:	Photomultiplier (PM2) R1450 <i>Hamamatsu</i> , Szintillator NE102a ($12 * 12 * 0.85\text{mm}^3$), Abstand Kathode/Szintillator 15cm .

3.2.2 Beschreibung

Man kann grob (allgemein, ohne Coulombkorrektur) das relative Betaspektrum der benutzten Quelle, als Überlagerung der beiden Einzelspektren (Strontium, Yttrium) berechnen. Aus

$$n(\epsilon)d\epsilon = C\epsilon\sqrt{\epsilon^2 - 1}(\epsilon_o - \epsilon)^2d\epsilon$$

mit $\epsilon = \frac{E_{ges}}{m_e c^2}$ und $\epsilon_o = \text{Maximalenergie}$, ergibt sich die Verteilung der Abbildung 3.11 (a), so normiert, daß die Wahrscheinlichkeit ein Elektron zu finden, 1 beträgt

²¹Die durch den Kollimator begrenzte Rate bedingte einen mittleren zeitlichen Abstand von etwa 10^{-4}sec zwischen zwei registrierten Ereignissen. Die Signalbreite war mit etwa 10^{-8}sec wesentlich kleiner.

(d.h. die Fläche unter der Kurve auf 1 normiert ist)²². [MY68]

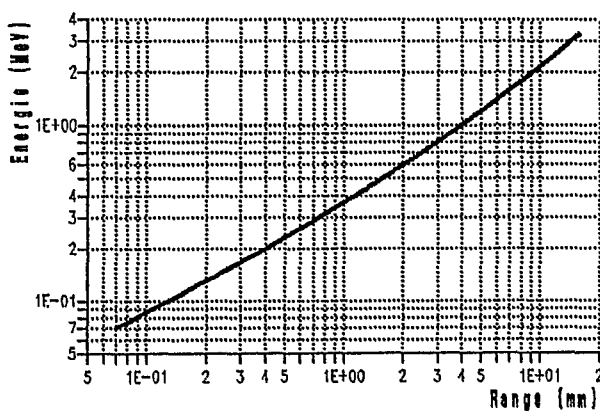
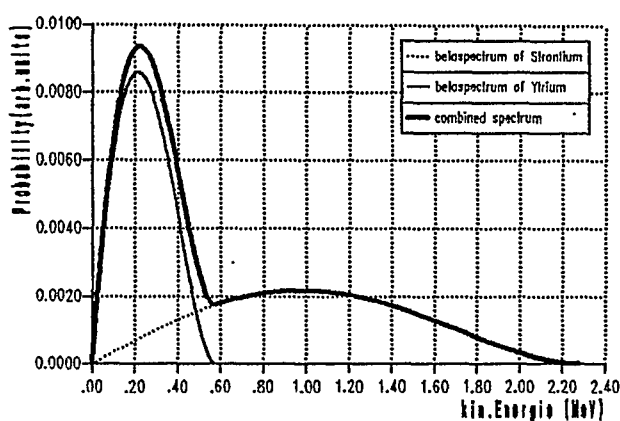


Abbildung 3.11: (a) Betaspektrum für Strontium/Yttrium (b) Reichweite von Elektronen in Szintillatormaterial

Auf der anderen Seite läßt sich die Reichweite²³ der Elektronen in Szintillatormaterial aus der empirischen Beziehung von Katz und Penfold ableiten zu: [MY68]

$$R\left(\frac{mg}{cm^2}\right) = 412[E(MeV)]^n,$$

$n = 1.265 - 0.0954 \ln[E(MeV)]$, für $0.01 \leq E(MeV) \leq 3$ (Abbildung 3.13 (b))

Um in SC1 ein Signal zu erzeugen, müssen die Elektronen sowohl den 'Quirl' als auch den Szintillator von SC2 durchqueren. Mit dieser Strecke von mindestens $3.85mm$ bekommt man aus obiger Reichweitebeziehung die Wahrscheinlichkeit, mit der ein solches Elektron bestimmter Energie ein GATE auslösen kann:

$$P(s) = \exp(-3.85\rho_{med}/R)$$

Faltet man die beiden Wahrscheinlichkeiten miteinander, so erhält man das Energiespektrum der Elektronen, die 'Events' erzeugt haben. Diese Verteilung gibt Figur 3.12 wieder.

²²Die Konstante C ist ein Maß für die Kernmatrixelemente und die Kopplungskonstanten. Da hier nur die relativen Häufigkeiten von Interesse waren, wurde C so gewählt, daß sich die gewünschte Normierung ergab.

²³Die Strecke nach der noch $1/e$ der Elektronen im Strahl vorhanden sind.

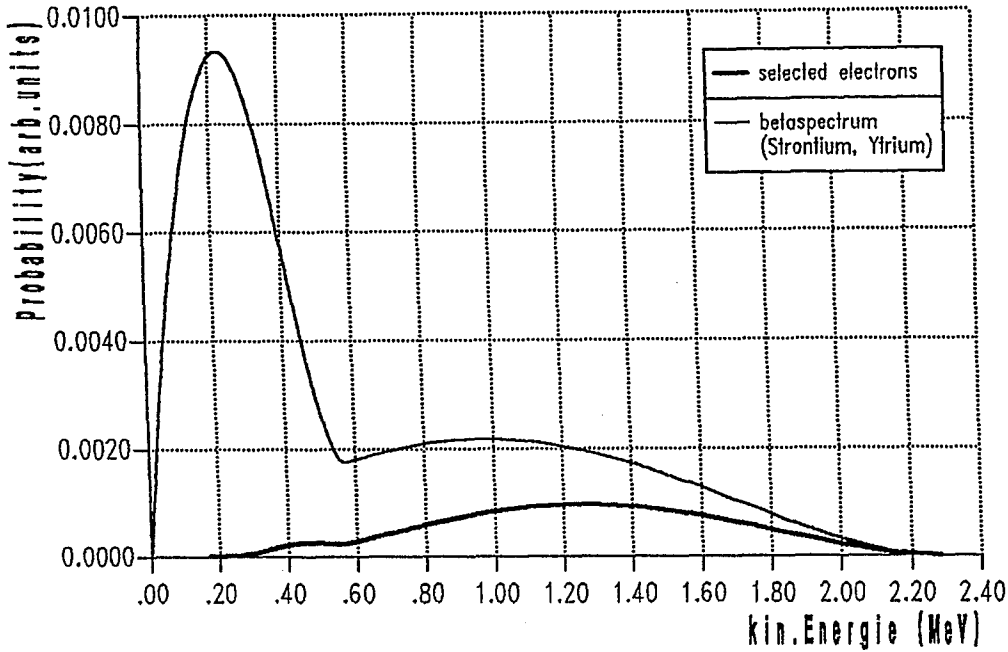


Abbildung 3.12: Vergleich zwischen Energieverteilung der vom Trigger selektierten Elektronen und Betaspektrum (Strontium/Yttrium)

Da für den späteren Einsatz der ungünstigste Fall abgeschätzt werden sollte, wurde der 'Quirl' mit Teilchen bestrahlt, die möglichst geringe Lichtausbeute ergaben. Alle relativistischen, ionisierenden Teilchen besitzen für die im Mittel pro Länge des durchquerten Materials deponierte Energie, bei einem β von etwa 0.96, ein Minimum mit $\frac{dE}{dx} \approx 2.3 \frac{\text{MeV}}{\text{cm}}$ (MINIMAL IONISIERENDE TEILCHEN). [Fer86] Die Beziehung zwischen Energieverlust ($\frac{dE}{dx}$) und kinetischer Energie der Projektile in Szintillatormaterial, ist für die verwendeten Elektronen in Abbildung 3.13 dargestellt²⁴. [Bet30] [Fer86]

²⁴Die Kurve gibt die wahrscheinlichste Energieabgabe pro Länge wieder und wurde nach der Formel:

$$\frac{dE}{dx} = D_e \left(\frac{Z_{in}}{\beta_{in}} \right)^2 n_c \left(\ln \left(\frac{2m_e c^2 \beta^2 \gamma^2}{I} \right) - \beta^2 \right)$$

(Bethe-Bloch Formel) mit $D_e = \frac{4\pi e^4}{m_e c^2}$, $n_e = \rho_{med} \frac{Z_{med}}{A_{med}} N_A$, $I = 16 Z_{med}^{0.9}$, $\rho_{med} = 1.032 \frac{\text{g}}{\text{cm}^3}$, $N_A = \text{Avogadrokonstante}$ berechnet. In dem betrachteten Energieintervall konnte man die Korrekturterme für Dichte- und Schaleffekte vernachlässigen.

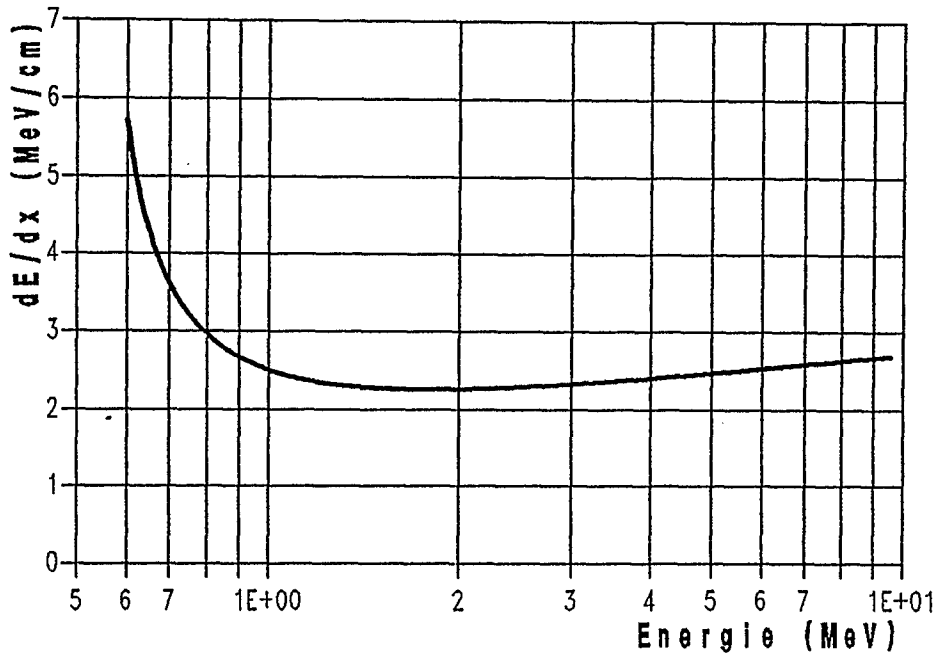


Abbildung 3.13: Wahrscheinlichster Enerverlust für Elektronen in Szintillatormaterial aufgetragen gegen ihre kinetische Energie (Bethe-Bloch).

An dieser Stelle sei auf das nicht zu vernachlässigende 'Straggling' (multiple scattering) der sehr leichten Projektile hingewiesen.

Wenn ein geladenes Teilchen ein Material durchquert, wird sein Weg durch viele elastische Stöße²⁵ beeinflusst ('*drunken mans way*'). Die Verteilung seiner Richtung nach einer Strecke x im Medium weicht zu größeren Winkeln von einer Gaußverteilung um die Ursprungsrichtung ab und kann mit Hilfe der Moliertheorie genauer beschrieben werden. Ein Fit an diese Theorie ergibt die Halbwertsbreite α_0 :

$$\alpha_0 = \frac{14.1 \text{ MeV}/c}{p\beta} Z_{in} \sqrt{\frac{L}{L_R}} \left[1 + \frac{1}{9} \log_{10} \left(\frac{L}{L_R} \right) \right] (\text{radians})$$

mit L/L_R = Dicke in Strahlungslängen und β, p, Z_{in} gleich Geschwindigkeit, Impuls und Ladung des Teilchens.

²⁵Für die benutzten Elektronen nur durch Coulombstreuung.

Im Bereich $10^{-3} < \frac{L}{L_R} < 10$ ('Quirl'+Szintillator: $\frac{L}{L_R} \approx 10^{-2}$) liefert diese Näherung Ergebnisse mit einem Fehler von 5%, der auf 10-20% für langsame Projektile und sehr leichte Elemente anwächst. Figur 3.14 zeigt die Abhängigkeit dieser Breite α_0 von der Energie der Teilchen.

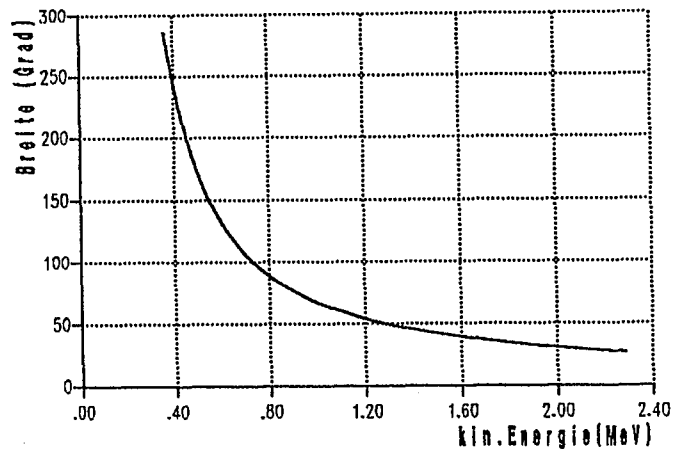


Abbildung 3.14: Abhängigkeit des 'stragglings' der Elektronen von ihrer kinetischen Energie.

Ein minimal ionisierendes Teilchen würde im 'Quirl' im Mittel eine Energie von 0.7 MeV deponieren. Aus obigen Betrachtungen folgt für die Elektronen der benutzten Strontiumquelle, daß deren Energieverlust maximal um einen Faktor 2 über dem Wert der minimal ionisierenden Teilchen liegen kann²⁶.

3.3 Messung

Das Photo auf der nächsten Seite zeigt den gesamten Versuchsaufbau und seine Anordnung in der lichtdicht verschließbaren Kiste (ohne Trigger). Für beide Konfigurationen (37°/51°) wurden analog je 22 Positionen, wie in Abbildung 3.1 skizziert, gemessen²⁷. Die exakten Koordinaten (Koordinatensystem wie in Kapitel 2.2 eingeführt) und die benutzte Nomenklatur sind:

- vom Zentrum nach Außen: E → A
(längs des 'Quirls')
- von der inneren zur äußeren Randkurve: 1 → 5
(quer zum 'Quirl')

²⁶Für Elektronen mit einer kinetischen Energie bis 1.6 MeV - sie verlieren im unteren Trigger-szintillator etwa 0.2 MeV - gilt dies sowieso. Elektronen mit höherer Energie erfahren dafür ein geringeres 'Stragglings'.

²⁷Zur Beschreibung wird im folgenden nur noch auf die erste Konfiguration Bezug genommen. Angaben zur 2. Konfiguration werden wenn in Klammern beigelegt.

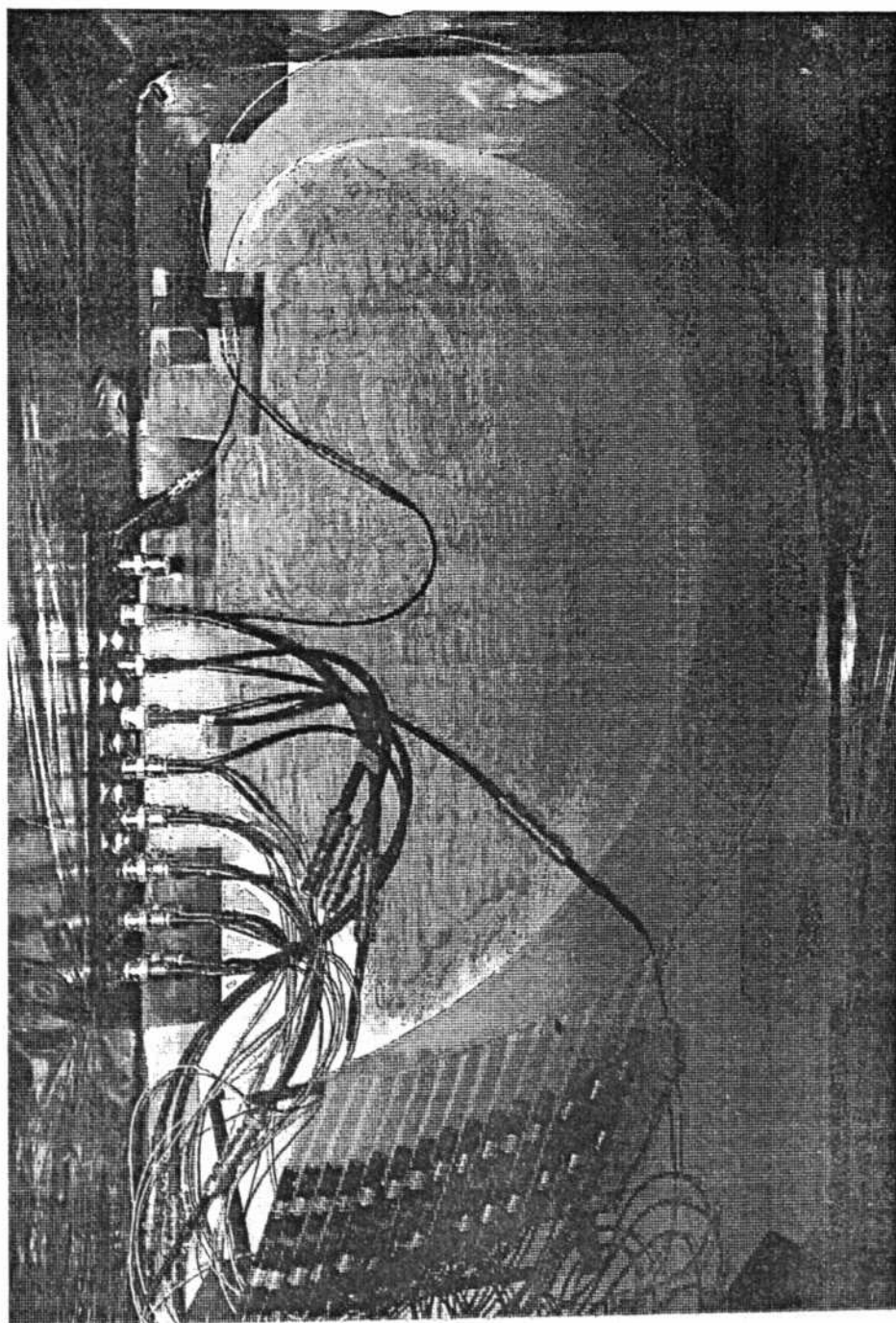


Abbildung 3.15: Das Photo zeigt den gesamten Versuchsaufbau in der schwarz ausgekleideten Aluminiumkiste (ohne Trigger). Links oben, an der Spitze des 'Quirls', ist die Leuchtdiode zu sehen, die zum Eichen der Photomultiplier benutzt wurde.

$$A = \begin{pmatrix} -37.7/13.6 \\ -38.8/14.6 \\ -39.8/15.7 \\ -40.9/16.6 \\ -42.2/17.7 \end{pmatrix}, B = \begin{pmatrix} -25.9/22.4 \\ -26.4/23.9 \\ -27.0/25.2 \\ -27.7/26.6 \\ -28.1/27.5 \end{pmatrix}, C = \begin{pmatrix} -6.8/24.2 \\ -7.3/25.8 \\ -7.7/27.0 \\ -7.0/28.4 \\ -6.3/30.0 \end{pmatrix},$$

$$D = \begin{pmatrix} 5.0/16.1 \\ 6.0/16.4 \\ 6.8/16.9 \\ 7.7/17.3 \end{pmatrix}, E = \begin{pmatrix} 4.7/2.4 \\ 5.3/1.7 \\ 6.0/1.3 \end{pmatrix}$$

Maße in [cm]; der Fehler in der Positionierung wird mit $\pm 3mm$ angenommen.
Die Datenaufnahme erfolgte mit dem Programm **ACQUIRE** (siehe Anhang C).

3.3.1 Eichung

Die Photomultiplier-Vorverstärker, die zur Verfügung standen, erlaubten kein ablesbares Einstellen des Verstärkungsfaktors. Um Baseline und relative Verstärkung zu protokollieren, wurde vor und nach den Meßpositionen (A - E) entlang dem 'Quirl' eine Eichmessung durchgeführt. Dazu gab man auf alle Kanäle parallel eine Folge von Referenzsignalen²⁸. Abbildung 3.16 (a) zeigt ein solches Spektrum. Daraus erhält man eine Eichgerade (siehe Figur 3.16 (b)), deren reziproke Steigung proportional der Verstärkung ist.

Während der gesamten Dauer der Messung diente ein freier ADC- und Verstärkerkanal dazu, einen Aussage über die Stabilität des Meßsystems (inclusive Pulser) zu erhalten. Bei festem Verstärkungsfaktor nahm man hierzu jedesmal die Referenzsignale mit auf. Aus dieser Meßreihe ergibt sich für Mittelwert und Standardabweichung

$$\begin{aligned} \text{Verstärkung} &: 20.7 \pm 0.14 \quad \left[\frac{\text{Kanäle}}{\text{rel. Amplitude}} \right] \\ \text{Baseline} &: 49.1 \pm 1.1 \quad [\text{Kanäle}] \end{aligned}$$

Die Schwankung der Verstärkung lag also, über die gesamte Meßdauer betrachtet deutlich unter 1%.

²⁸Die Normpulse erzeugte ein schneller Pulser (Phillips PM5786B, 1Hz-125MHz, risetime 2nsec-0.1sec), der 4 definiert wählbare Ausgangsamplituden im Verhältnis 5:2.5:1:0.4 (und 0) lieferte. Dieses Signal wurde dann bis zu 16fach aufgeteilt und über die Verstärker auf die ADC's gegeben.

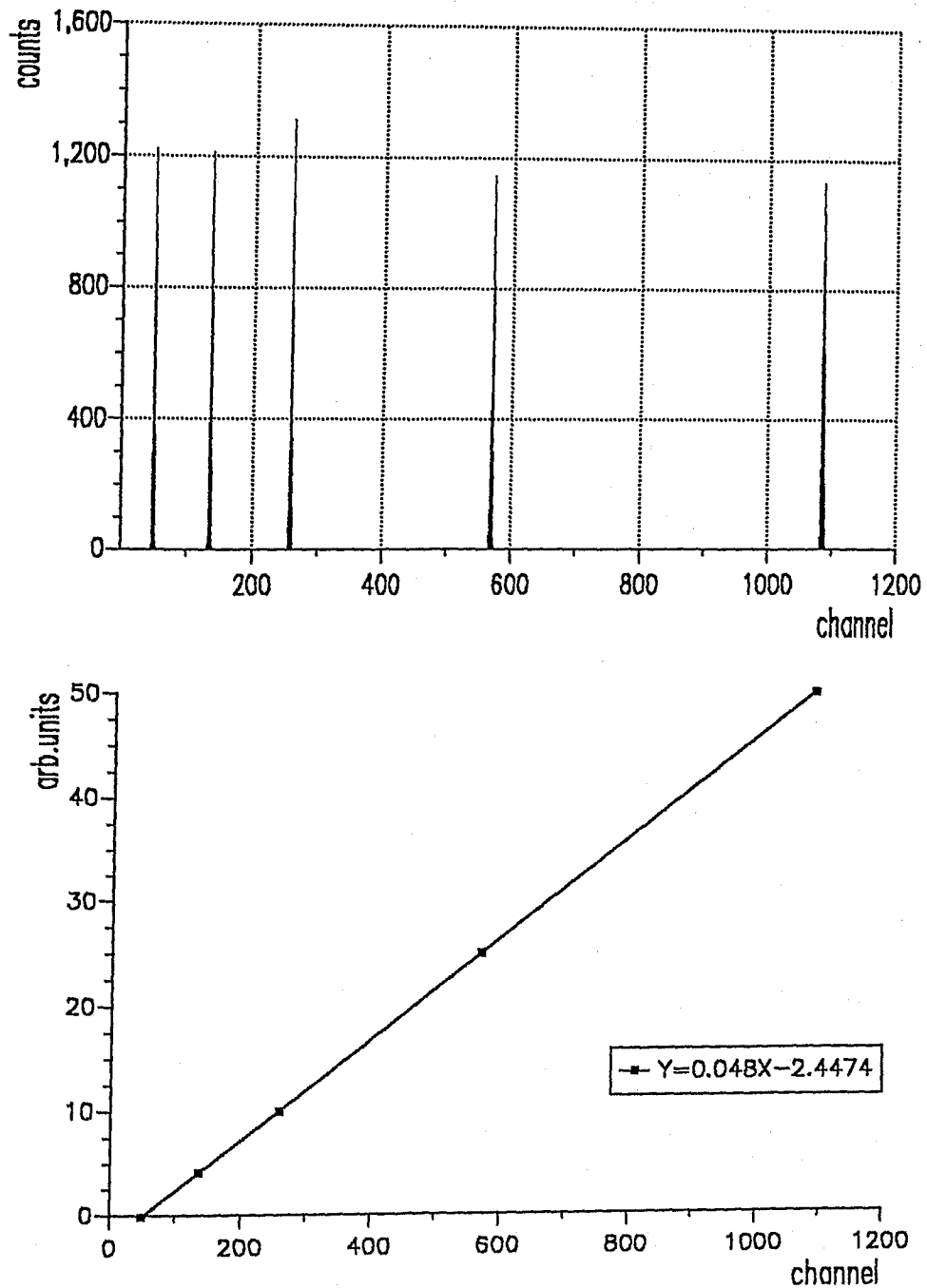


Abbildung 3.16: (a) Spektrum der Referenzsignale - die Halbwertsbreite beträgt etwa 3 Kanäle; (b) Eichgerade deren reziproke Steigung proportional der eingestellten Verstärkung ist.

Nach Beendigung der Messung wurden mit Hilfe einer gepulsten Leuchtdiode die Photomultiplier geeicht²⁹. Man schwächt hierzu das Pulssignal³⁰, welches die Diode zum Leuchten bringt so weit ab, bis nur noch maximal ein Photon pro Puls die Photomultiplierkathode erreicht. Auch hierbei wurde der Verstärkungsfaktor, wie oben beschrieben mitprotokolliert.

Die Leuchtdiode war so angebracht, daß sie im Zentrum in den 'Quirl' einstrahlte³¹ und sich alle 11 (15) Photomultiplier auf einmal eichen ließen (siehe Abbildung 3.15).

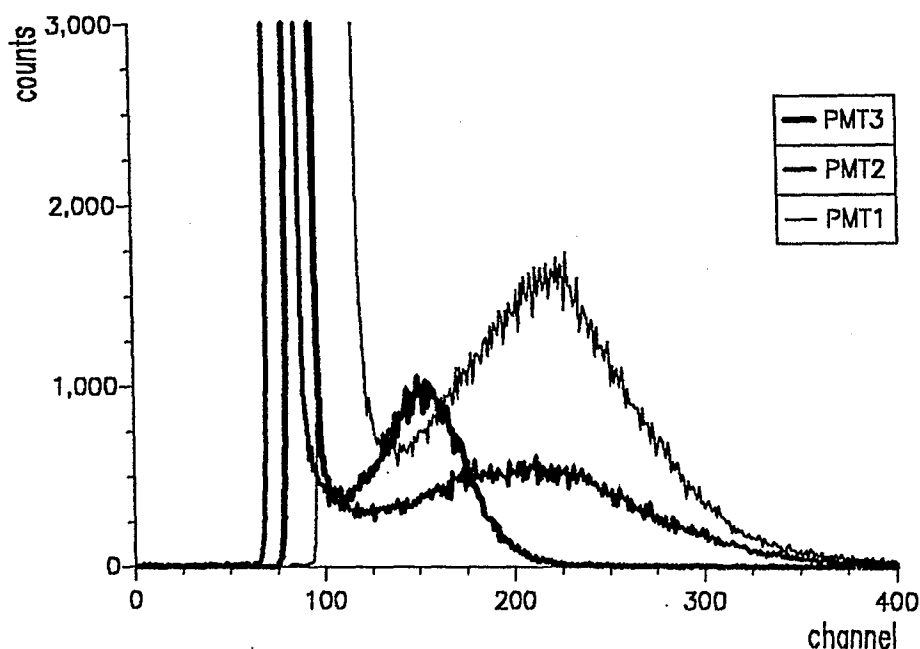


Abbildung 3.17: Dargestellt sind 3 der zum Eichen der Photomultiplier aufgenommenen 'single electron peaks'.

Mit Hilfe des so aufgenommenen 'singel electron' Signals (SER)³² konnten die Amplitudenspektren in Vielfache dieses 'Ein-Elektronen-Ereignisses' umgerechnet werden.

²⁹Die Photomultiplier wurden konstant mit der vom Hersteller empfohlenen Hochspannung betrieben ($-1250V$).

³⁰Benutzt wurde ein im IKP weiterentwickelter Pulser, der speziell zur Simulation von schnellen Szintillatorsignalen durch Leuchtdioden geeignet ist. [Sef88]

³¹Das hatte weiter den Vorteil, daß die starke Einstrahlung des Pulsers in die Photomultiplier (durch die Entfernung Diode - PM) fast vollständig verschwand.

³²SER: single electron response, daß Signal das man am Ausgang eines Photomultipliers erhält, wenn genau ein Elektron aus der Kathode befreit und bis zur Anode vervielfältigt wird.

Dies geschieht nach der Beziehung:

$$A_{ser} = g \frac{A_{pos}}{SER_{pos}}$$

Hierbei bedeuten A_{ser} die Amplitude in Einheiten des 'single electron respons' - peaks, $g = \text{Verstärkung}_{(SER)} / \text{Verstärkung}_{(A)}$ das Verhältnis der beiden gemessenen, relativen Verstärkungsfaktoren (siehe oben) und SER_{pos} bzw. A_{pos} , die um die Baseline korrigierten Positionen des 'single electron respons' und der umzurechnenden Amplitude.

Die Position der SER-peaks wurde durch anfitzen einer Gaußfunktion an die Kurven (Figur 3.17) ermittelt³³. Betrachtet man die Standardabweichungen dieser Fits, so erhält man den relativen Fehler zu $\frac{\Delta SER_{pos}}{SER_{pos}} \lesssim 10\%$. Unter Berücksichtigung dieses Ergebnisses und der Stabilitätsmessung (Drift der Verstärkung und der Baseline) folgt aus dem Fehlerfortpflanzungsgesetz als Abschätzung für den Gesamtfehler:

$$\frac{\Delta A_{ser}}{A_{ser}} = 12\%$$

3.3.2 Meßergebnisse

Die Bezeichnung der Photomultiplier richtet sich nach ihrer Position auf Stirnfäche - am äußeren Rand PMT1, am inneren PMT11 (PMT15) - siehe auch Figur 3.1. Vornehmlich sollen die Meßpunkte bei E betrachtet werden, da sie die ungünstigsten Ergebnisse erwarten lassen und im späteren Experiment die innere Region des 'Quirls' besonders stark bestrahlt werden wird.

In Abbildung 3.18 sind drei geeichte Spektren miteinander verglichen. Der 'Quirl' wurde hierbei in Position E2 bestrahlt und es werden die Photomultiplier PMT1, PMT5 und PMT10 betrachtet (Konfiguration 1).

Es wird ersichtlich, daß

- die mittlere Anzahl der pro Ereignis registrierten Photonen in PMT1, d.h. am äußeren Rand, am größten ist und nach innen hin abnimmt.

³³Die benutzte Routine entstammte dem HBOOK, einem vom CERN angebotenen Softwarepaket.

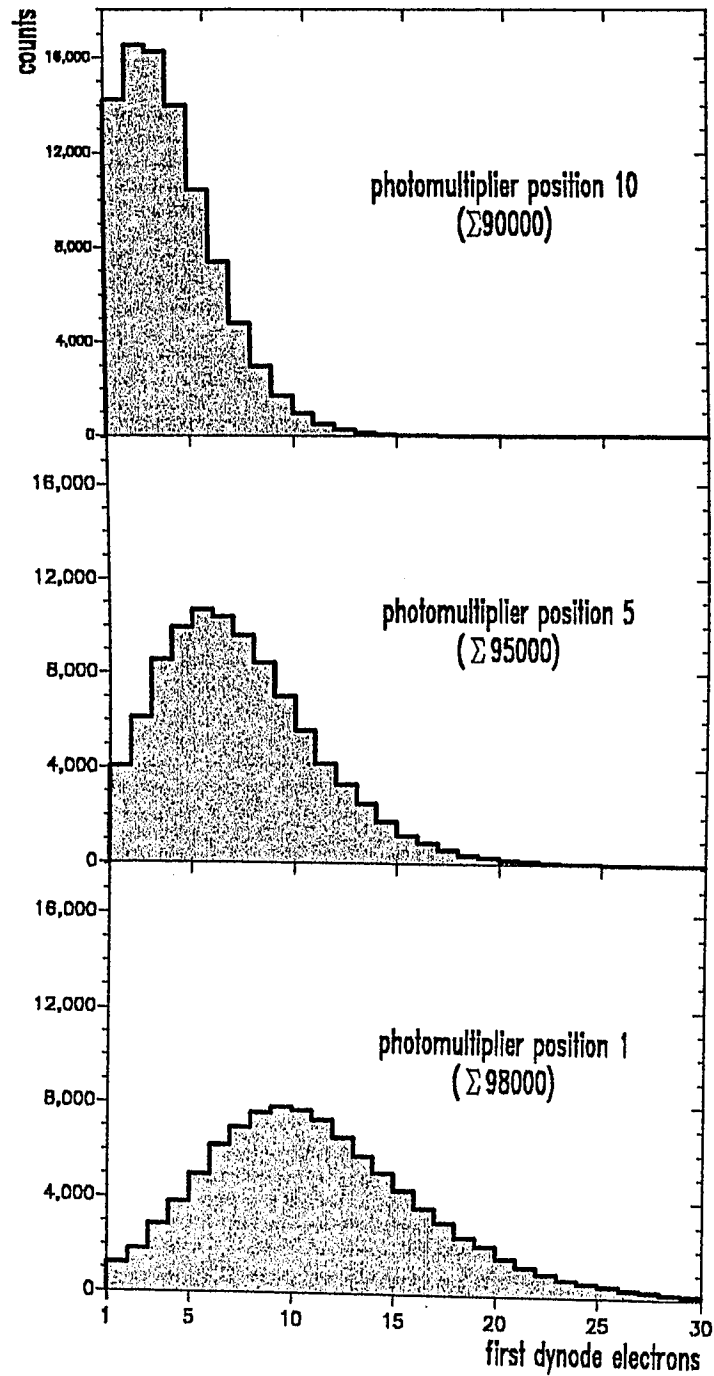


Abbildung 3.18: Vergleich dreier geeichteter Spektren: PMT1, PMT5 und PMT10 - Bestrahlungsort E2, 100 000 registrierte Ereignisse.

- die drei Photomultiplier in Abhängigkeit von ihrer Position in 90 – 98% der Ereignisse, wenigstens ein Photon registriert haben³⁴.

Die Erzeugung von Photoelektronen in der Kathode stellt einen statistischen Prozess dar, der einer Poissonverteilung gehorcht. Diese nähert sich für größere \bar{n} (mittlere Anzahl erzeugter Photoelektronen) einer Gaußverteilung an:

PMT10 → PMT1. Im weiteren wird zur Beschreibung der einzelnen Spektren die mittlere Anzahl, der pro Ereignis registrierten Photonen \bar{n} benutzt.

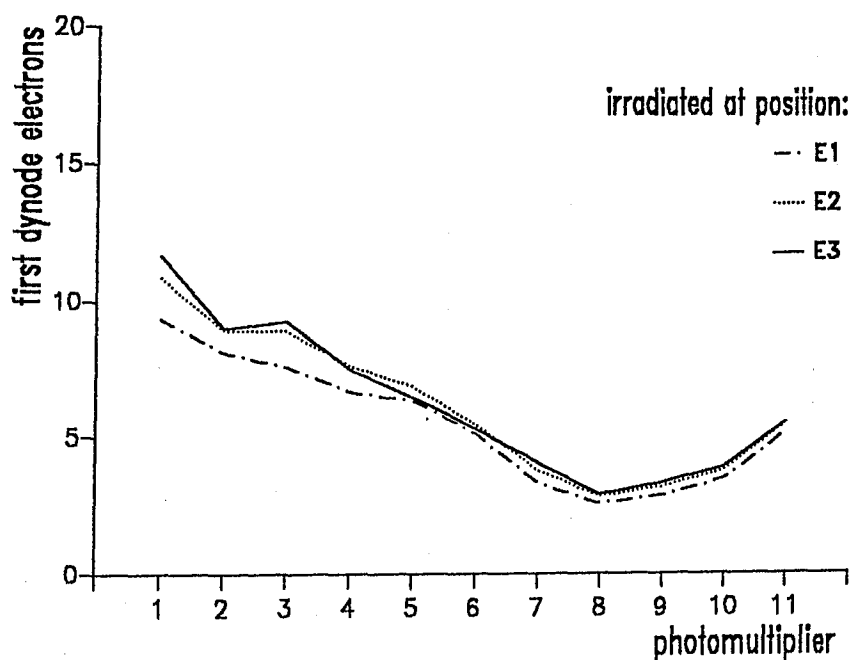


Abbildung 3.19: Mittlere Anzahl der registrierten Photonen pro Ereignis, aufgetragen gegen die Photomultiplierposition auf der Stirnfläche - Bestrahlungsorte E_i .

Die ungleichmäßige Ausleuchtung der 'Quirl'-Stirnfläche zeigt auch das Diagramm 3.19 Hier sind die Mittelwerte der registrierten Photonen gegen deren jeweilige Position auf der Stirnfläche, für die 3 Bestrahlungsorte E1, E2 und E3 aufgetragen. Die Lichtausbeute steigt stetig mit den Photomultipliern zum äußeren Rand hin an. Der

³⁴Die Ursache für die Ereignisse, bei denen kein Photon gemessen wurde, liegt bei der Poissonverteilung, die eine von 0 verschiedene Wahrscheinlichkeit für 'Nullereignisse' besitzt. Das Minimum der registrierten Ereignisse liegt für die Meßposition E2 mit 82% bei PMT8.

kleinere Anstieg für die Positionen 9 - 11 ist durch die Art der Lichtauskopplung überbetont³⁵. Man erkennt eine nur schwache Abhängigkeit vom Bestrahlungsort quer zum 'Quirl'; für den inneren Punkt E1 ergibt sich eine etwas geringere Ausbeute.

Betrachtet man die anderen Meßpositionen längs des 'Quirls' ($D_i - A_i$), so wird der Anstieg zum äußeren Rand hin stärker und die Abhängigkeit der Lichtausbeute von der Position der Elektronenquelle quer zum 'Quirl' nimmt zu; beispielhaft³⁶ zeigt dies ein Vergleich zwischen Diagramm 3.19 und 3.20.

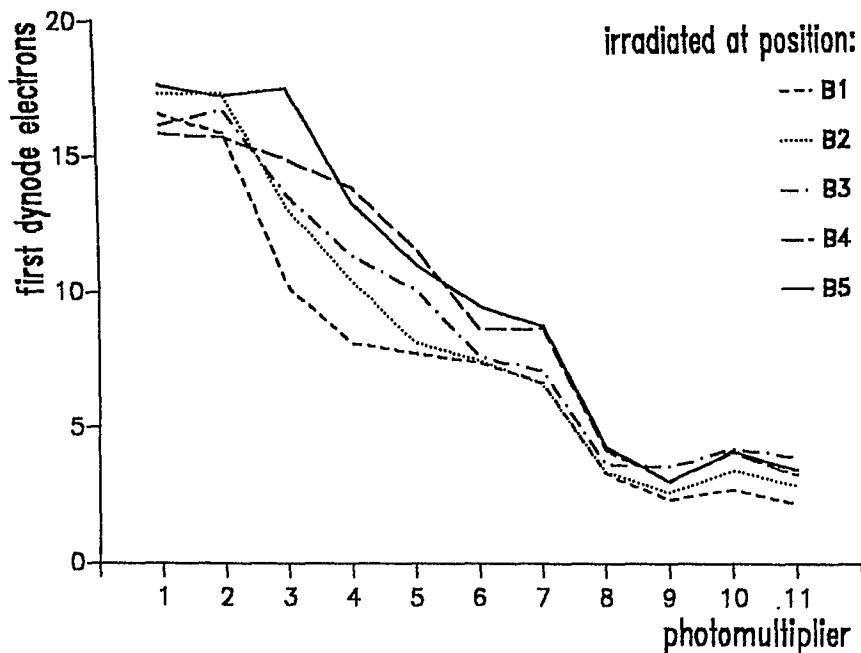


Abbildung 3.20: Mittlere registrierte Photonenzahl pro Ereignis aufgetragen gegen die Multiplierposition - Bestrahlungsorte B_i .

Bei der Datenaufnahme wurden neben den Spektren auch die einzelnen 'Events'

³⁵An den 'Leerstellen' zwischen den angesetzten Plexiglaslichtleitern können die Photonen verstärkt in den 'Quirl' zurückreflektiert werden. Die Form der inneren Spirale lenkt sie dann in die Spitze, also zu den Positionen 9 - 11 hin. Der Effekt tritt besonders für die Meßpunkte E_i auf, da die Photonen (wie die Simulation zeigte) fast ausschließlich 'an der äußeren Spirale entlang laufen', so daß sie im Mittel einen recht großen Winkel zur Normalen der Stirnfläche aufweisen und eine Reflexion wahrscheinlicher wird.

³⁶Die Diagramme zu den übrigen Meßpunkten sind in Anhang E wiedergegeben.

- als Vektor mit allen Inhalten der ADC- und TDC-Kanäle pro Ereignis - weggeschrieben. Damit war man in der Lage, 'offline' integrale Spektren (Summation über die 11 (15) Photomultiplier) zu erstellen, die eine erste Näherung für die spätere Meßanordnung³⁷ ergeben. Figur 3.21 zeigt das integrale Spektrum für den Meßpunkt E2.

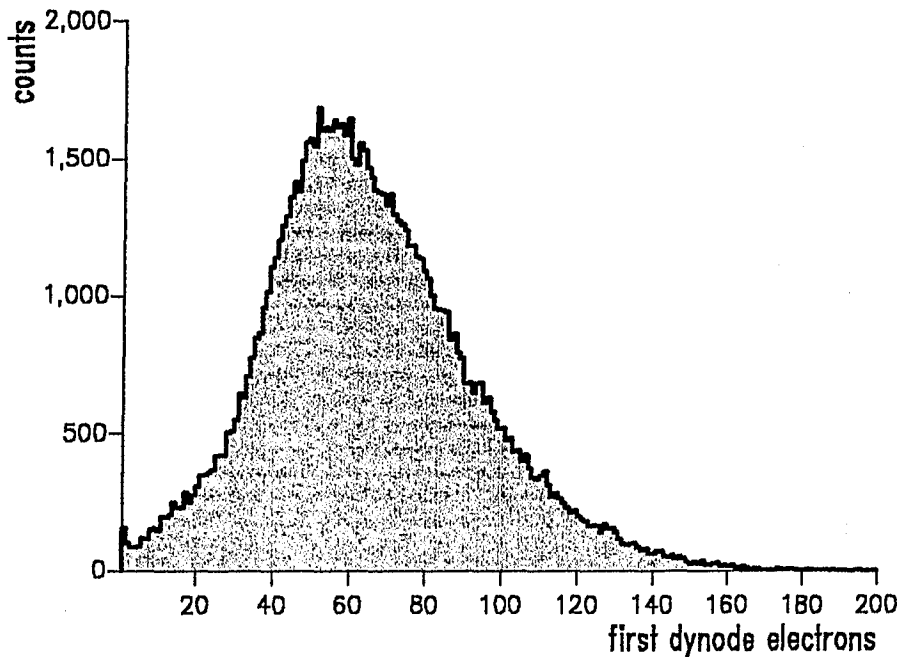


Abbildung 3.21: Integrale Lichtausbeute des 'Quirls': Bestrahlungsort E2, summiert über die 11 Photomultiplier.

Als Mittelwert erhält man für die Position E2 etwa $\bar{n} = 65$ nachgewiesene Photonen. Dieser Wert muß noch auf die Gesamtstirnfläche normiert werden³⁸, womit sich $\bar{n} = 78$ ergibt.

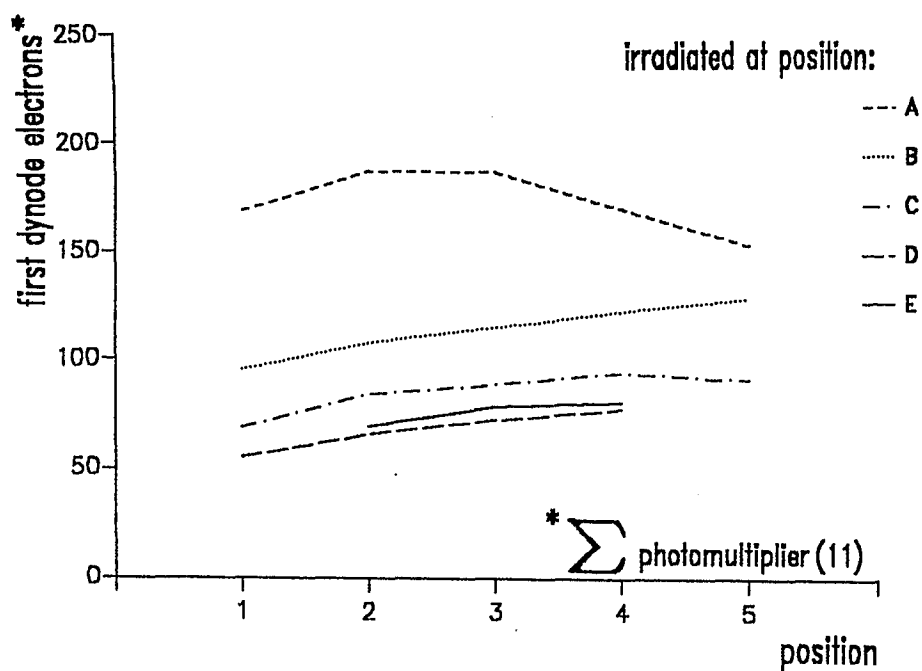
Diese normierten Mittelwerte der Lichtausbeute des 'Quirls' sind in den beiden fol-

³⁷Die Lichtauskopplung soll dann durch einen einzelnen Lichtleiter erfolgen, der die gesamte 'Quirl'-Stirnfläche abdeckt und auf einen Photomultiplier mit entsprechend großer Kathodenfläche führt.

³⁸Hier muß das Verhältnis *optisch kontaktierte Fläche / Gesamtfläche*, welches 1 : 1.4 beträgt, berücksichtigt werden. Es wurde weiter angenommen, daß die Hälfte, der an den 'Leerstellen' zwischen den Plexiglas-Ansatzstücken reflektierten Photonen, trotzdem auf eine Photomultiplierkathode gelangten.

genden Abbildungen 3.22 (Konfiguration 1) und 3.23 (Konfiguration 2), für alle Meßpositionen ($A_i - E_i$) aufgetragen. Aus den Diagrammen ergibt sich:

- Die Konfiguration 1, das heißt ein Ansetzen der Lichtleiter unter einem Winkel von 37° zur Stirnfläche (siehe Abbildung 3.1.2), erwies sich wie erwartet als die bessere Lösung. Das Verhältnis der Lichtausbeute zu Konfiguration 2 beträgt etwa 4 : 3.
- Mit Konfiguration 1 wurden für den **ungünstigsten Fall** im Mittel rund **56 Photonen** pro Ereignis gemessen. Der ungünstigste Fall ergab sich nicht, wie zu erwarten wäre, für die Positionen E_i , sondern für die Positionen D_i ³⁹.
- Für die Messungen, bei denen der 'Quirl' im hinteren Bereich ($E_i - A_i$, mit Einschränkung auch B_i) bestrahlt wurde, erkennt man eine nur geringe Abhängigkeit der Lichtausbeute von der Position längs des 'Quirls'⁴⁰



Abbildungung 3.22: Mittelwerte der integralen Lichtausbeute des 'Quirls' pro Ereignis für alle Meßpositionen: Konfiguration 1.

³⁹Eine Erklärung dafür ist die in der Nähe des Zentrums stärkere, relative Verbreiterung des Lichtleiters (schön zu sehen an dem 360° -Quirl' auf Seite 35), was zu den guten optischen Eigenschaften für die Bestrahlungspunkte E_i führt ('aufgehende Geometrie').

⁴⁰Diese geringe Abhängigkeit von der Position längs des 'Quirls', das 'Plateau', wurde aufgrund der Simulationsrechnungen erwartet.

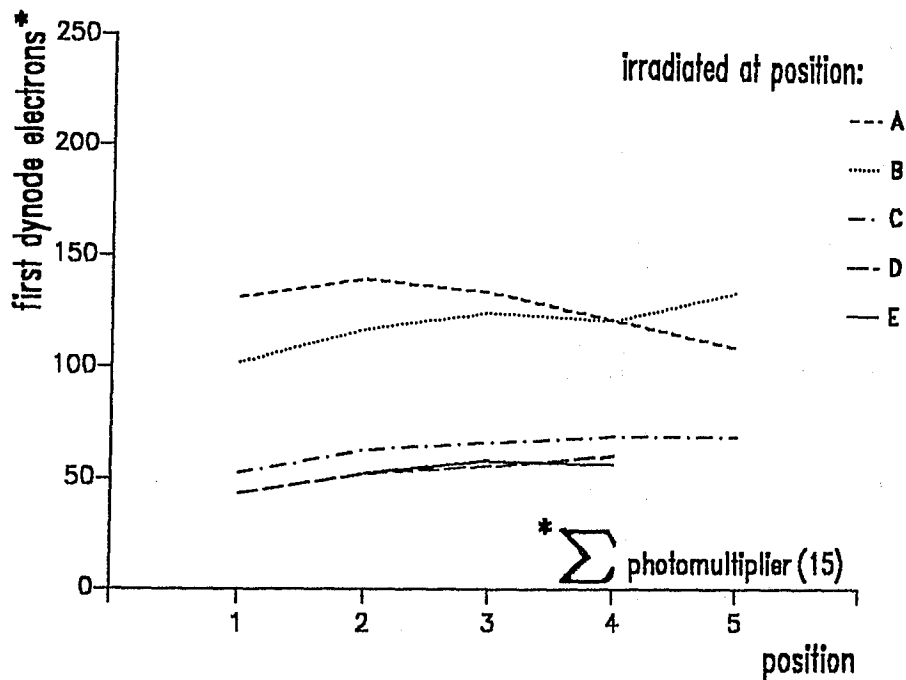


Abbildung 3.23: Mittelwerte der integralen Lichtausbeute des 'Quirls' pro Ereignis für alle Meßpositionen: Konfiguration 2. (Die Positionen 1 - 5 liegen quer zum 'Quirl', A - E entlang dem 'Quirl'.)

Aus den aufgenommenen TDC-Spektren sollen Aussagen über das Zeitverhalten des 'Quirl' abgeleitet werden. Es standen 8 TDC-Kanäle zur Verfügung; die mit TDC's verbundenen Photomultiplier waren:

Konfiguration1	1/ 2/ 3/ 5/ 7/ 9/10/11
Konfiguration2	1/ 3/ 5/ 7/ 9/11/13/15

In Abbildung 3.24 sind die Zeitspektren für 3 Photomultiplier wiedergegeben - Bestrahlungsort: C1; Photomultiplier: PM1, PM7, PM11.

Da die Erzeugung der Photonen, deren Weg im 'Quirl' und ihre Umwandlung in ein elektrisches Signal statistische Prozesse darstellen, erwartet man, daß (für viele registrierte Ereignisse: $N \rightarrow \infty$) ⁴¹ die Zeitspektren gaußverteilt sind.

⁴¹ Zentraler Grenzwertsatz der Wahrscheinlichkeitsrechnung. Auf die Form der Spektren soll später genauer eingegangen werden.

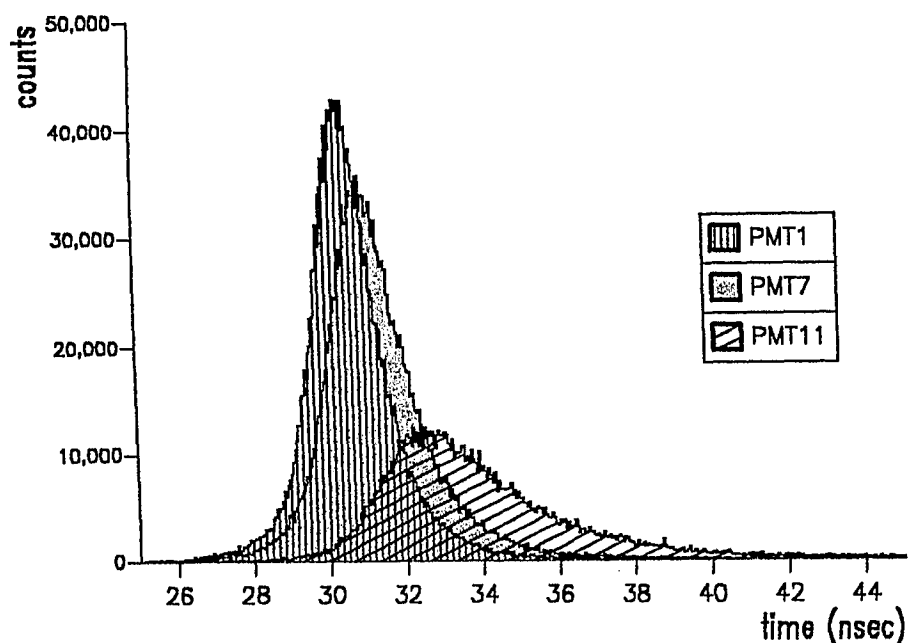


Abbildung 3.24: Vergleich der Zeitspektren dreier Photomultiplier, Bestrahlungsort: C1. Die Zeitskala besitzt einen relativen Zeitnullpunkt.

- Man erkennt, daß die Photonen den Photomultiplier am äußeren Rand (PMT1) im Mittel am schnellsten erreichen. Dies gilt mit Ausnahme einiger Bestrahlungsorte in der Nähe der Stirnfläche (A_i) für alle Messungen.

Die einzelnen Spektren sollen durch ihre charakteristischen Größen - Position des Maximums (mittlere Laufzeit der Photonen im 'Quirl'), Halbwertsbreite (Laufzeit-'jitter') - beschrieben werden. Diagramm 3.25 zeigt diese für 4 Meßpositionen: A1, B1, C3 und D2⁴². Es wird hier die Position des Maximums gegen die Photomultiplierposition aufgetragen; der relative Nullpunkt der Zeitskala ist identisch mit dem in Abbildung 3.24.

- Für einige Bestrahlungsorte ergeben sich in Abhängigkeit der Photomultiplierposition auf der Stirnfläche Laufzeitdifferenzen (Position der Maxima) von mehr als 2 nsec .

⁴²Die Diagramme zu allen Messungen mit Konfiguration 1 sind in Anhang E zusammengestellt.

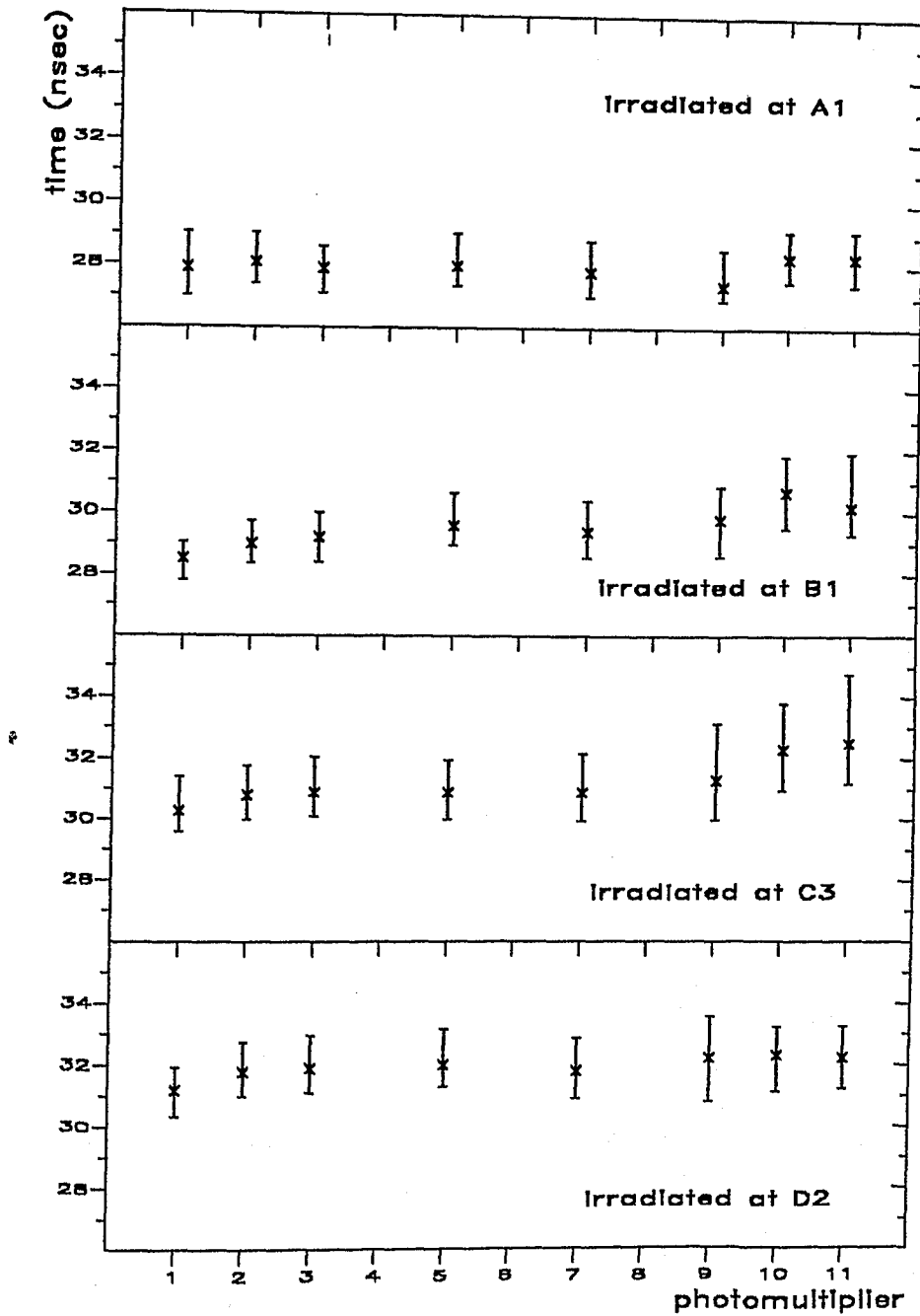


Abbildung 3.25: Mittlere Laufzeit der Photonen im 'Quirl', aufgetragen gegen die Photomultiplierpositionen.

Die Bogenlänge der Spirale berechnet sich zu⁴³:

$$L(\varphi) = \frac{c}{2} (\varphi \sqrt{1 + \varphi^2} + \ln(\varphi + \sqrt{1 + \varphi^2}))$$

mit $c = \frac{r_{max}}{\varphi_{max}}$.

Die Bestrahlungsorte A₃, B₃, C₃, D₂ und E₂ liegen etwa auf einer Spirale, die die Mitte zwischen den beiden Randkurven bestimmt. Trägt man für diese die Laufzeit gegen die Bogenlänge bis zum jeweiligen Bestrahlungspunkt auf ($L_{max} - L(\varphi)$, Nullpunkt an der Stirnfläche), so ergibt sich eine Gerade:

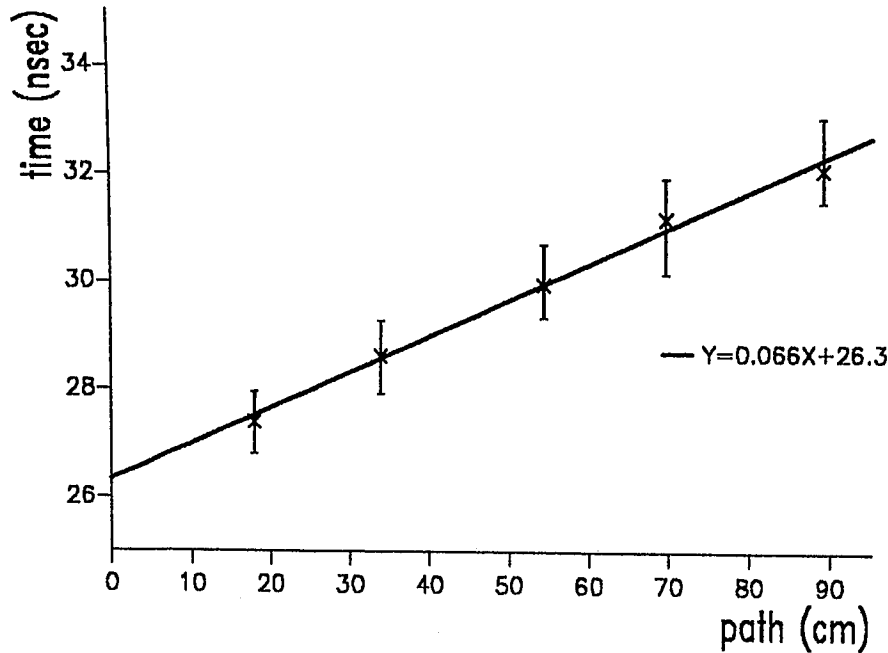


Abbildung 3.26: Weg-Laufzeit-Diagramm für die Meßpunkte, die auf der 'Mittelspirale' liegen (A₃/B₃/C₃/D₂/E₂).

- Die maximale Laufzeit des Lichts im 'Quirl' beträgt etwa 6nsec.
- Durch Extrapolation kann man damit die relative Zeitskala eichen:

$$t_o = 26.3 \pm 0.75 \text{ nsec}$$

⁴³Die Funktion der Spirale im R₂ lautet: $f(\varphi) = (c\varphi \cos \varphi, c\varphi \sin \varphi)$. Für die Bogenlänge einer Kurve gilt: $L(s) = \int_0^s |f'(t)| dt$ und mit $f'(\varphi) = (c \cos \varphi - \varphi c \sin \varphi, c \sin \varphi + \varphi c \cos \varphi)$ folgt: $L(s) = \int_0^s \sqrt{1 + t^2} dt$. Die Lösung für dieses bestimmte Integral findet man in [BS84].

Schließlich betrachtet man die Zeitauflösung, die mit dem 'Quirl' in diesem Versuchsaufbau erreicht wird. In Diagramm 3.27 ist für PMT1 die Zeitauflösung gegen die zugehörige mittlere Anzahl \bar{n} der registrierten Photonen (genauer: gegen $\frac{1}{\sqrt{\bar{n}}}$) aufgetragen⁴⁴.

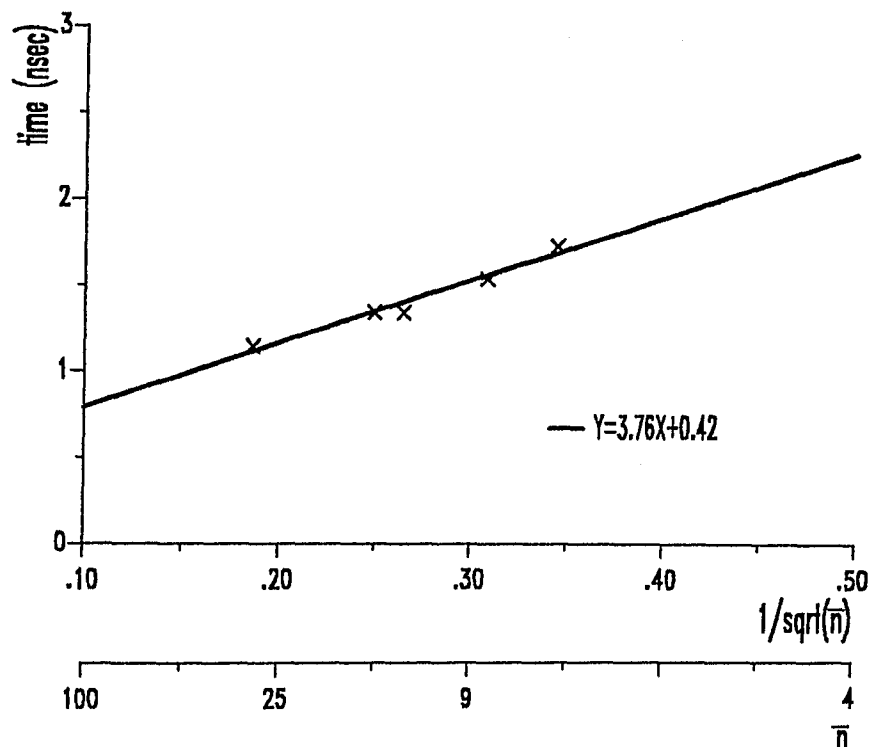


Abbildung 3.27: Zusammenhang zwischen Zeitauflösung und mittlerer Anzahl registrierter Photonen \bar{n} für PMT1.

- Die Zeitauflösung ist hier zur reziproken Wurzel aus \bar{n} proportional: $\Delta t \sim \frac{1}{\sqrt{\bar{n}}}$. Für PMT1 und die betrachteten Meßpunkte liegt die Zeitauflösung zwischen 1.1 und 1.6 nsec.

Im späteren Experiment werden alle ausgekoppelten Photonen auf einen Photomultiplier geführt $\rightarrow \bar{n}$ ist damit größer. In erster Näherung wurde die Gerade in Abbildung 3.27 extrapoliert und für 50 – 60 Photonen im Mittel eine Zeitauflösung von $\Delta t = 1 \text{ nsec}$ ermittelt⁴⁵.

⁴⁴Der relative Fehler in der Bestimmung von \bar{n} beträgt 12%. Die zweite Skala ordnet \bar{n} zu. Es wurden die 5 Meßpunkte auf der 'Mittelspirale' betrachtet.

⁴⁵Die so bestimmte Zeitauflösung Δt setzt sich aus verschiedenen Komponenten zusammen: • Δt_{trig} der durch den Trigger erzeugte Zeit-'jitter'. Eine Kontrollmessung ergab für die benutzte Einstellung $\Delta t_{\text{trig}} = 0.75 \pm 0.1 \text{ nsec}$. • Δt_{PM} Zeit-'jitter' des Photomultipliers (hier PMT1). Eine

Zur Form der Zeitspektren:

Die Diagramme 3.24 und 3.25 zeigen, daß manche Zeitspektren nicht nur wegen der geringeren Zahl, der im Mittel gemessenen Photonen eine größere Halbwertsbreite besitzen, sondern zu längeren Laufzeiten hin von der zu erwartenden Gaußverteilung abweichen. In diesen Fällen ist die Wahrscheinlichkeit der Photonen, den jeweiligen Photomultiplier zu erreichen so gering, daß die Photonen, die nach einer Reflexion an der Stirnfläche registriert werden verstärkt ins Gewicht fallen. Es ergeben sich damit zwei Gaußverteilungen für zwei mittlere Laufzeiten, die sich überlagern.

Messung von Ralph Thomas mit dem gleichen Photomultipliertyp R1635-02 (im Rahmen seiner Diplomarbeit) lieferte $\Delta t_{PM} = 0.35 nsec$ für $\bar{n} = 60$ (relativer Fehler 15%). • Δt_Q Zeit-'jitter' aufgrund der Laufzeitunterschiede der Photonen im 'Quirl'. Eine erste Näherung ergibt $\Delta t_Q = \sqrt{\Delta t^2 - \Delta t_{PM}^2 - \Delta t_{trig}^2}$ mit $\Delta t_Q \approx 0.6 nsec$.

Kapitel 4

Zusammenfassung

Das zentrale Ergebnis dieser Arbeit lautet:

- Der 'Quirl' mit seiner speziellen Geometrie stellt einen Szintillator mit guten **Lichtleitereigenschaften** dar.

Dies erscheint bei seiner Form zunächst überraschend, aber weitaus die meisten Reflexionen der Photonen finden (analog zum Tortenstück) wegen der, gegenüber den anderen Ausmaßen geringe Dicke, an den planen Flächen ('Boden' und 'Deckel') und nicht an den Seiten statt. Auch besitzt der 'Quirl' eine aufgehende Geometrie, die zu den guten Eigenschaften (zum Beispiel eine weniger divergente Winkelverteilung als rechteckige Lichtleiter) führt.

Die Messung ergab für die zur Bestrahlung des 'Quirls' benutzten Elektronen im ungünstigsten Fall (D1) eine Lichtausbeute von knapp 60 Photonen.

- Um eine gute Lichtauskopplung zu erzielen, sollte die Stirnfläche senkrecht zur tangentialen Verlängerung der Außenspirale sein. Dabei kommt es im wesentlichen auf den äußeren Bereich an.

Diese Tangente repräsentiert die mittlere Photonrichtung an der Stirnfläche und ein Großteil der Photonen, besonders die zeitlich frühen, erreichen den empfohlenen Bereich¹.

¹Für die Stirnfläche ist ein Kompromiß gesucht, da die 3 Szintillatorlagen möglichst vollständig überlappen und im Ganzen eine Keisfläche bilden sollen.

- Parallel zur tangentialen Verlängerung der äußeren Randkurve angesetzten Lichtleitern, ist - wie erwartet - der Vorzug zu geben.

Dies zeigte die Kontrollmessung (Konfiguration 2). Damit wird die Lichtauskopplung gleichzeitig für die in der Nähe des Zentrums erzeugten Photonen optimiert².

Der Gesamtaufbau des JETSET-Detektorsystems erfordert zur Auslese des dreilagigen Szintillator-Hodoskops längere, speziell geformte Lichtleiter, welche zu weiteren Verlusten führen werden. Die Zeitauflösung gehorcht der Beziehung:

$$\Delta t = \frac{1}{\sqrt{n}}$$

(\bar{n} = mittlere Anzahl nachgewiesener Photonen). Unter Berücksichtigung dieser Tatsachen wurde die Dicke der 'Quirl'-Segmente (obwohl ein möglichst 'masseloser Detektor angestrebt wird) auf 5mm erhöht.

Mit der Dicke von 5mm folgt in Kombination aus Mess- und Simulationsergebnissen für den ungünstigsten Fall eine mittlere Ausbeute von rund 100 Photonen.

- Der Vergleich *Software* (OPAQUE) \leftrightarrow *Experiment* zeigt gute Übereinstimmung.

Mit dem Programm hat man die Möglichkeit, interessierende Eigenschaften für Variationen des 'Quirls' oder andere Körper, als erste Abschätzung, zu berechnen. Ein Beispiel dafür ist das aus der Simulationsrechnung zu einem Tortenstück erhaltene Verhältnis:

$$\frac{\text{Ausbeute 'Quirl'}}{\text{Ausbeute Tortenstück}} \approx \frac{2}{3}$$

- Eine grobe Abschätzung der Zeitauflösung des 'Quirls' liefert:
 $\Delta t_Q \approx 0.6nsec.$

Die Abschätzung ist nur eingeschränkt auf die späteren Versuchsbedingungen (1 Photomultiplier pro 'Quirl'-Segment) übertragbar. Um die Zeitauflösung genauer zu bestimmen, ist ein weiterer Testaufbau, dann unter echten Bedingungen, vorgesehen.

²Neben der insgesamt höheren Lichtausbeute für Konfiguration 1, ist auch die Differenz, der im Mittel nachgewiesenen Photonen zwischen E_i und A_i kleiner. Man möchte auf diese innen erzeugten Photonen optimieren, da die Anzahl der registrierten Photonen für die äußeren Messpunkte unkritisch ist.

An dieser Stelle möchte ich mich für die ausgezeichnete Betreuung und das großzügige Infrastrukturangebot der KFA (Rechenzentrum, mechanische und elektronische Werkstätten) bedanken. Zu betonen ist das hervorragende Arbeitsklima innerhalb der Gruppe der Mittelernergiephysik.

Im Einzelnen gilt mein besonderer Dank

- Herrn Prof. Dr. K. Kilian, der die Arbeit am Institut für Kernphysik der Kernforschungsanlage Jülich ermöglichte und zahlreiche fachliche Anregungen gab.
- Herrn Dr. habil. W. Oelert, für die direkte Betreuung dieser Arbeit, die umsichtige Leitung unserer Arbeitsgruppe und seinen aufmunternden Beistand.
- Herrn A. Wasserziehr, für seine Hilfe, sowohl bei mechanischen Arbeiten als auch auf der Großrechenanlage.
- Herrn H. Hadamek (Leiter der mechanischen Werkstatt des IKP) für sein Verständnis gegenüber '*Physikern*' und sein reges Interesse an der Bearbeitung von Szintillatoren und Lichtleitern.
- Den anderen Mitgliedern der Mittelernergie-Gruppe und allen die zum Gelingen dieser Diplomarbeit beigetragen haben.

Meinen Eltern und meinen Münchnern

Kapitel 5

Anhang

A · Ergänzung zu den verwendeten PMTs

Beim Kauf erhält man von HAMAMATSU zu jedem Photomultiplier individuelle Informationen aus einem Abschlußtest. Beispielhaft sollen diese für drei Röhren des Typs R1635-02 angegeben werden¹.

Serial Number	RX6034	RX6043	RX6056
Cathode Luminous Sens. $\mu A/lm$	110.0	107.0	114.0
Anode Luminous Sens. A/lm	134.0	237.0	109.0
Anode Dark Current nA	3.88	0.55	0.38
Cathode Blue Sens. $\mu A/lm(blue)$	10.30	10.50	10.60

Notes

- (1)(2)(4) Light source: Tungsten filament lamp operated at 2856K.
- (2)(3) Overall supply voltage: 1250V, standart socket.
- (3) The bulb of the tube is insulated from ground potetial.
- (4) Measured with a Corning CS 5-58 blue filter (half stock thickness)

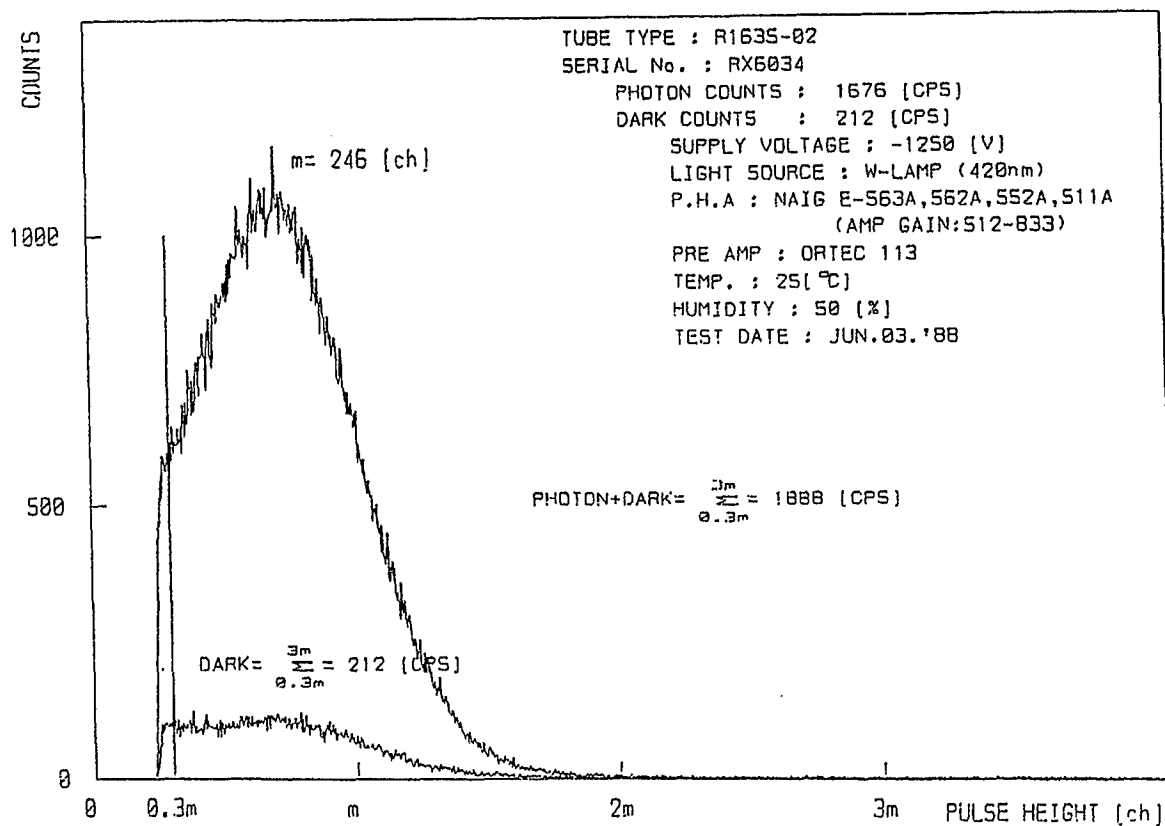
Date: June 10,1988 Temperature: 25°

Die folgenden Testspektren zeigen sehr deutlich das gute Rauschverhalten und sind ein Beispiel für ein 'Ein-Elektronen-Ereignis'² (SER, single electron response).

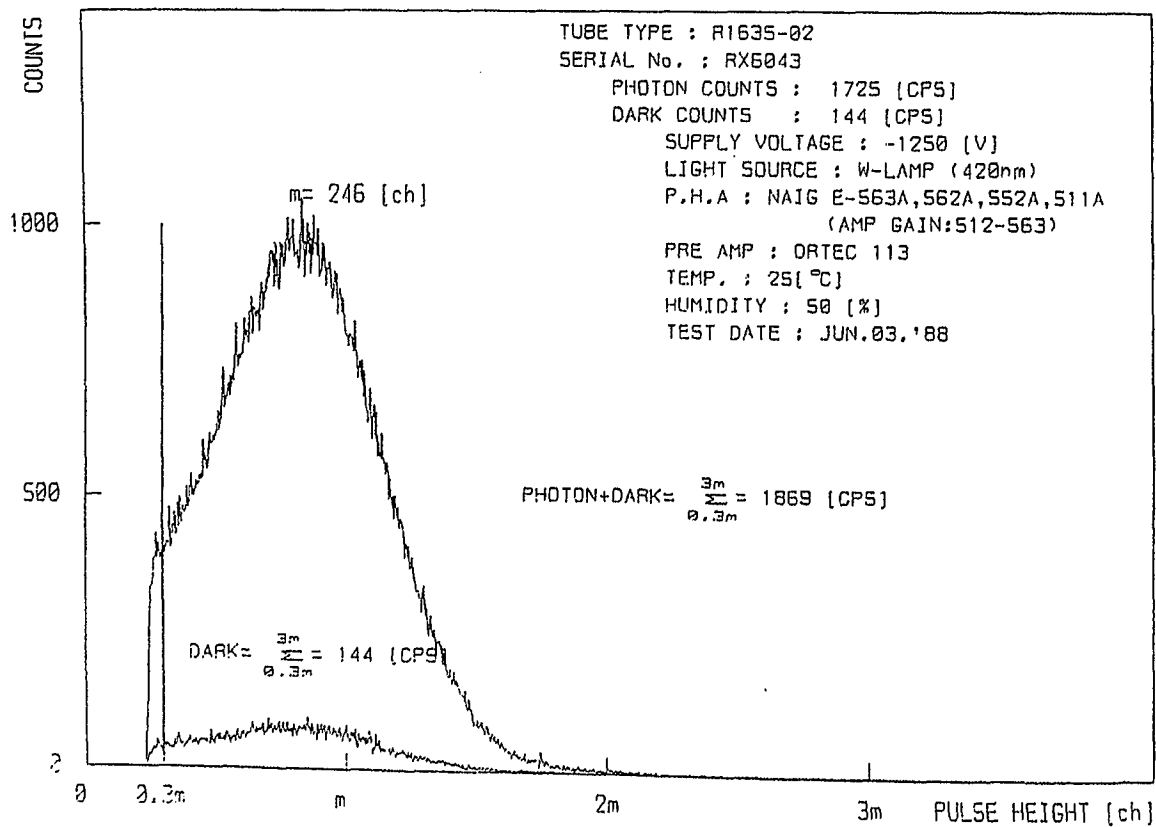
¹Die übrigen verwendeten Photomultiplier liegen, analog der RX6043, mit ihren Werten zwischen den beiden Extremen RX6034 und RX6056

²Da die Wahrscheinlichkeit dafür, daß ein aus der Kathode befreites Photoelektron die erste Dynode erreicht, der Überföhrungsfaktor, heute in der Regel sehr nahe bei 1 liegt, wurden in dieser Arbeit 'Erste Dynode Elektron' und Photoelektron synonym benutzt.

SINGLE PHOTOELECTRON P.H.D



SINGLE PHOTOELECTRON P.H.D



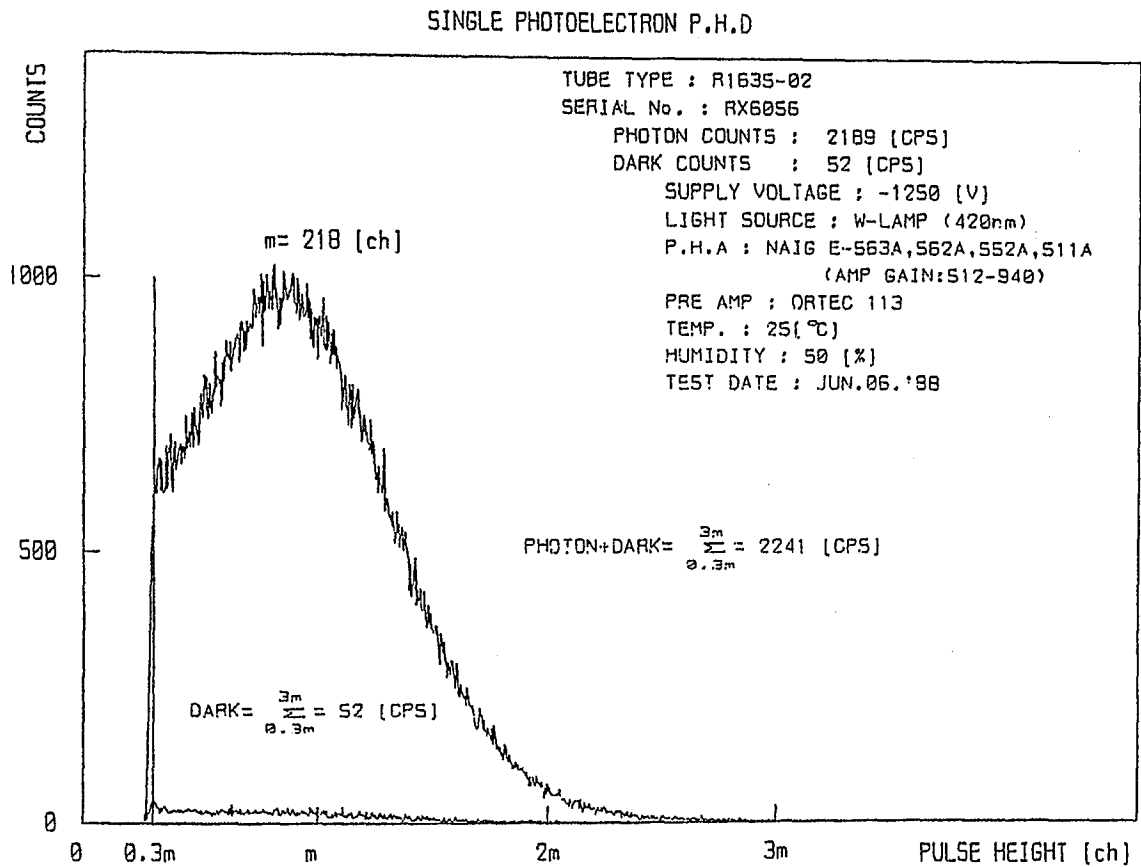


Abbildung 5.1: Testspektren dreier Photomultiplier R1635-02

Im Experiment wurden die Stirnflächen mit maximal 15 Röhren bestückt. Ihre Seriennummern lauten:

RX6099	RX6086	RX6082
RX6104	RX6081	RX6043
RX6021	RX6013	RX6017
RX6034	RX6078	RX6030
RX6059	RX6056	RX6074

[HAM88a]

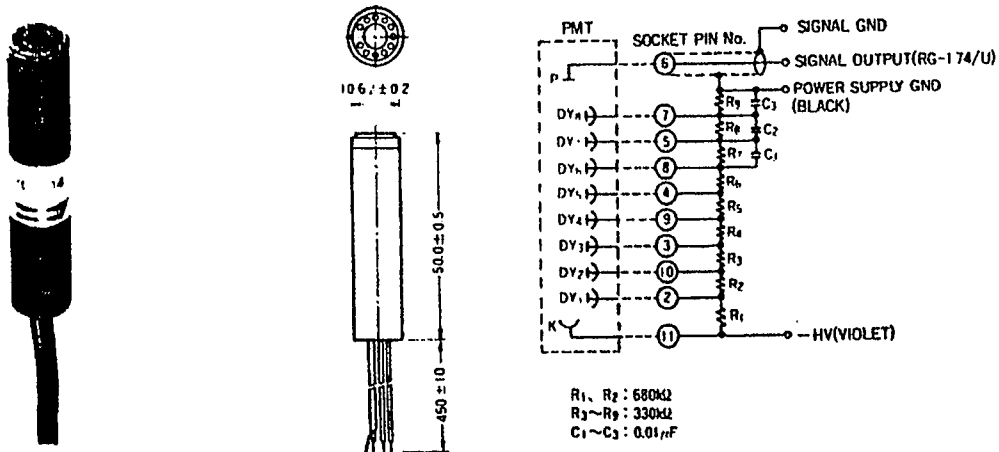


Abbildung 5.2: Base E1761-04 mit Schaltbild [HAM86]

Verwendet wurden Basen von HAMAMATSU (E1761-04), die zu den Photomultipliern R1635-02 angeboten werden und deren Spannungsteiler an diese angepaßt sind. Abbildung 5.2 zeigt die Base und das Schaltbild des Spannungsteilers.

B Ergänzung zur Elektronik (ACQUIRE)

- (i) Ergänzende Daten zu den benutzten Elektronikmodulen
- (ii) Listing des Datenaufnahmeprogramms ACQUIRE

(i)

DISC	NIM Modul 821 Quad Discriminator (LeCroy 1983)
High Sensitivity:	$-30mV$
Output width:	$5nsec$ to $1\mu sec$
Maximum Rate:	$110MHz$ typical, input and output
Double-Pulse Resolution:	Less than $9nsec$
Input Impedance:	50Ω , NIM level
Risetime:	$< 2.0nsec$
Falltime:	$< 2.5nsec$
COIN	NIM Modul 364AL Dual four-fold majority logic gate with veto (LeCroy 1980)
Input Impedance:	50Ω , NIM level
Veto Input:	05Ω , NIM level
Duration:	fixed $3.8nsec$ or equal to time overlap
Risetime/Falltime:	$1.2nsec$ typical
Coincidence width:	$1nsec$ up, determined by input pulse durations
Rate:	$150MHz$ minimum
Double Pulse Resolution:	typical $5nsec$
AMP	NIM Modul 612AM 6 Channel Photomultiplier Amplifier
Input Impedance:	50Ω , NIM level
Gain:	2.5 to 40
Linearity:	0.2% integral
Coupling:	direct
Risetime:	$< 3nsec$
Noise:	Less than $50\mu V$ rms, referred to input, total

Tabelle 5.1: Daten zu den verwendeten LeCroy NIM-Modulen

delayed TRIGGER	NIM Modul DT103 Dr.B.Struck ³ , 3 independant channels
Input:	50 Ω , NIM level
Output:	NIM, TTL, inverse NIM
Delay:	10nsec – 10sec
Width:	5nsec – 10sec
Jitter:	over the whole range approx. 10 ⁻³

Tabelle 5.2: Daten zum delayed Trigger

Zu dem DSP⁴ Interface Computer-CAMAC gehören:

- Crate Controller & Dataway Display, Model DSP6001
- Microcomputer Interface, PC004
- Software-Einbindung in verschiedene Computersprachen (FORTRAN, PASCAL, Turbo-Pascal, ...)

(ii)

An dieser Stelle soll das Listing des Programms ACQUIRE wiedergegeben werden, welches zur Steuerung des CAMAC-Systems und zur Datenaufnahme benutzt wurde. Es ist in 'Turbo Pascal 3'⁵ auf einem IBM Personal Computer (At) erstellt worden und grob in folgende Files untergliedert :

- Acquire.PAS (Hauptprogramm) Initialisieren, Reservierung von Speicherplatz und Usermenue
- Camturbo.INC DSP-Interface, Kommunikation mit dem CAMAC-System
- Getting.ING Routinen zur Ein- und Ausgabe (Schirm/Tastatur)
- Graphiks.INC Erstellen der Menues, Fenster und Plots
- Select.INC Realisation der Optionen
- Collect.INC Datenaufnahme

³Dr.B.Struck, 2000 Düsseldorf-Tangstedt, Bäckerberg 6

⁴DSP Cooperation, 48500 Kato Road, Fermon, CA 94538-7338

⁵©Borland. Die Update Version 'Turbo Pascal 4' konnte leider keine Verwendung finden, da der Befehl EXTERNAL, mit dem die Assembler Routinen des 'DSP-Interface' eingebunden werden zu den wenigen gehört, welche nicht aufwärts kompatibel sind.

Die Laufparameter - Anzahl der ADC (ach+1) / TDC (tch+1) Kanäle, Events pro cycle, cycle pro Messung, Camac-Stationen - werden im Sourcetext eingestellt. Die Schleife REPEAT ... UNTIL durchläuft das Hauptmenue solange, bis man es mit 'E'xit verläßt.

(● Acquire.PAS)

```

program Acquire;
{////////////////////}
const
    ast=2; { ADC-station }
    tst=5; { TDC-station }
    ach=15; tch=7; { Anzahl ADC-/TDC-Kanaele }
    param=24; { Anzahl der Parameter eines 'Event-Vektors' }
    depth=1919; { Anzahl der Kanaele pro ADC/TDC }
    freq=100; { Anzahl der 'cycle' }
    cycle=1000; { Anzahl der aufgenommenen 'Events' bis zum 'refresh'
                  der Graphik }

type
    ADCspectra=array[0..ach,1..depth] of integer;
    ad=^ADCspectra;
    TDCspectra=array[0..tch,1..depth] of integer;
    td=^TDCspectra;
    Events=array[1..cycle,1..param] of integer;
    ev=^Events;
    names=string[60];

var
    adc:ad; tdc:td; event:ev;
    a:char;
    filvar:file of Events;
    filt:text;
    InString,filename,filename,txt:names;
    i,recs,adca,adce,tdca,tdce,show,cycles,hoehe,sca,sct:integer;
    continue,fixed,modeadc:boolean;
    sclag:integer;
    aoverflow,toverflow:array[0..param] of real;
    wort:char;
{-----}
{$I typedef.sys }
{$I graphix.sys }
{$I kernel.sys }
{$I windows.sys }
{$I camturbo.inc }
{$I getting.inc }
{$I graphiks.inc }
{$I select.inc }
{$I collect.inc }
{-----}
begin
    adce:=depth; tdce:=depth; adca:=1; tdca:=1; show:=0;
    continue:=false; fixed:=false; hoehe:=64;
    filename:='AcQ'; txt:='no information specified!';
    InitGraphic;
    SelectScreen(2);

```

```

Init_Plot;
ClearScreen;
SelectScreen(1);
Init_Crate;
: new(adc); new(tdc); new(event);           { Speichererweiterung ! }
Init_Arrays;
: REPEAT
  ClearScreen;
  RestoreWindow(Menu,0,0);
  wort:=getin;
  case wort of
    'c','C': Collect_Data;
    'f','F': Get_Filename;
    's','S': Store_Spectra;
    'r','R': Restore_Spectra;
    'b','B': begin
      wr(45,6,'Adc/Tdc ? (a):');
      wort:=getin;
      if wort in ['t','T'] then modeadc:=false
      else modeadc:=true;
      Big_Plot;
      end;
    'd','D': Clear;
    'p','P': Set_Parameters;
  end;
  UNTIL wort in ['e','E'];
: LeaveGraphic;
: end.

```

Mit den folgenden Definitionen werden die Prozeduren des DSP Interface eingebunden.

(• Camturbo.INC)

```

.....
{MODULE      :DSPTurbo.inc
PURPOSE      :CAMAC I/O drivers EXTERNAL defines
: REVISION HISTORY
12/26/85 miller created
7/17/86 Cleaned up and made official dla
:
TYPE long_int = ARRAY [0..1] OF INTEGER;
:
{CAMAC PROCEDURES}
PROCEDURE CANTURBO;          external 'camturbo.bin';
PROCEDURE CAMO (var N,F,A,D,Q,X : INTEGER);  external camturbo[0];
PROCEDURE CAMI (var N,F,A,D,Q,X : INTEGER);  external camturbo[6];
PROCEDURE CAML (VAR L : integer);            external camturbo[12];
PROCEDURE CAMCL (var N : INTEGER);           external camturbo[18];
PROCEDURE DMAO (var N,F,A,ARR,ER : INTEGER); external camturbo[24];
PROCEDURE DMAI (var N,F,A,ARR,ER : INTEGER); external camturbo[30];
PROCEDURE DMASET (var Crate : integer;
  var BytesPerXfer : integer;
  var QMode : integer;
  var NoOfXfers : integer); external camturbo[36];
PROCEDURE CRATE_SET (var C : INTEGER);        external camturbo[42];
PROCEDURE camo24 (VAR n,f,a : INTEGER;
  VAR data : long_int;

```

```

    VAR q,x    : INTEGER);      EXTERNAL camturbo[48];
PROCEDURE cami24 (VAR n,f,a : INTEGER;
    VAR data   : long_int;
    VAR q,x    : INTEGER);      EXTERNAL camturbo[54];
PROCEDURE camcyc (VAR ncycles : INTEGER);      EXTERNAL camturbo[60];
PROCEDURE Real_cnvt (VAR IEEE_in : long_int;
    VAR real_out : REAL);      EXTERNAL camturbo[66];
:

```

Diese Unterroutrinen dienen der Ausgabe von Text in die Gaphikumgebung, in welcher die Menues und die Plotfenster programmiert sind.

(• Getting.INC)

```

:
var code:integer;
function getin:char;
var wort:char;
begin
    repeat until keypressed;
    read(kbd,wort);
    getin:=wort;
end;
{////////////////////////////////////}
function convert(inte,mLength:integer):names;
var i,iLength:integer;
begin
    str(inte,InString);
    iLength:=length(InString);
    if iLength<mLength then
        for i:=iLength+1 to mLength do begin
            InString:='_'+InString;
        end;
    convert:=InString;
end;
{////////////////////////////////////}
:
procedure wr(x,y:integer;txt:names);
begin
    Gotoxy(x,y);
    write(txt);
end;
{////////////////////////////////////}
function getarg(a,b,nr:integer):integer;
var i,j,erg :integer;
    inp:array[1..10] of integer;
    ready:boolean;
    wort:char; cont:names;
begin
    InString:='';
    for i:=1 to nr do inp[i]:=0;
    i:=0; ready:=true; erg:=0;
    repeat
        wort:=getin;
        ready:=(wort in ['0'..'9']);
        if ready then begin
            i:=i+1;

```

```

        InString:=InString+wort;
        wr(a+nr-i,b,InString);
        end;
    until (not ready) or (i=nr);
    val(InString,erg,code);
    if InString='' then erg:=-1;
    getarg:=erg;
    end;
{////////////////////////////////////}
function getstring(a,b,n:integer):names;
var i,j:integer;
    name:names;
begin
    name:='';
    i:=0;
    repeat
        wort:=getin;
        if wort in ['a'..'z','A'..'Z','0'..'9','.',',','-','_',' ' ] then begin
            name:=name + wort;
            wr(a+i,b,wort);
            i:=i+1;
        end;
    until (i=n) or not (wort in ['a'..'z','A'..'Z','0'..'9','.',',','-','_',' ']);
    getstring:=name;
end;
;

```

Hier finden sich die Unterprogramme, welche sowohl die Menüs realisieren als auch die Spektren (auch 'online' zur Verfolgung der Messung) erzeugen. Dabei kann man über das Panel 'PARAMETERS' auf die Form der Spektren einwirken.

(• Graphiks.INC)

```

;
const
    ADCs=1; TDCs=2; Info=3; Menu=4; Single=5; Inlay=6;
var
    maxi,scal:integer;
;
procedure Init_Plot;
{-----}
begin
    SetHeaderOn;
    SetHeaderToBottom;
    DefineWindow(ADCs,10,186,42,314);
    DefineHeader(ADCs,'ADC-Spektrum');
    SelectWindow(ADCs);
    DrawBorder;
    StoreWindow(ADCs);
    DefineWindow(TDCs,50,186,82,314);
    DefineHeader(TDCs,'TDC-Spektrum');
    SelectWindow(TDCs);
    DrawBorder;
    StoreWindow(TDCs);
    SetHeaderToTop;
    DefineWindow(Info,10,16,82,160);
    DefineHeader(Info,'Informations on ADC/TDC');
    SelectWindow(Info);
    DrawBorder;

```

```

StoreWindow(Info);
DefineWindow(Menu,10,16,82,160);
DefineHeader(Menu,'Menu');
SelectWindow(Menu);
DrawBorder;
wr(11,3,'WatE 5.11/88');
wr(24,5,'Filename:      F');
wr(24,6,'Parameters:    P');
wr(24,7,'Collect:       C');
wr(24,8,'Store Spectra: S');
wr(24,9,'Restore:       R');
wr(24,10,'Order:         0');
wr(24,11,'Exit:         E');
wr(44,5,'single Big Plot: B');
StoreWindow(Menu);
ClearScreen;
DefineWindow(Single,8,40,85,310);
DefineHeader(Single,'Einzelspektrum');
SelectWindow(Single);
DrawBorder;
StoreWindow(Single);
ClearScreen;
DefineWindow(Inlay,62,0,87,80);
DefineHeader(Inlay,'Info');
DefineWorld(4,0,0,38,16);
SelectWindow(Inlay);
DrawBorder;
StoreWindow(Inlay);
end;
:
procedure BIG_Plot;
{-----}
var
  i,j,xw,yw:integer;
  mac,ch,cha,che:integer;
begin
  SelectScreen(2);
  ClearScreen;
  maxi:=0; scal:=hoehe;
  if modeadc then begin
    ch:=ach; cha:=adca; che:=adce;  scflag:=ADCs;
    for i:=cha to che do
      if (adc^[show,i]>maxi) and (i<1919) then begin
        maxi:=adc^[show,i];
        mac:=i;
      end;
    end;
  else begin
    ch:=tch; cha:=tdca; che:=tdce;  scflag:=TDCs;
    for i:=cha to che do
      if (tdc^[show,i]>maxi) and (i<1919) then begin
        maxi:=tdc^[show,i];
        mac:=i;
      end;
    end;
  end;
  if not fixed then while scal<maxi do scal:=scal*2;
  RestoreWindow(Single,0,0);
  DefineWorld(3,cha,0,che,scal);
  SelectWorld(3);
  SelectWindow(Single);
  for xw:=cha to che do begin
    if modeadc then yw:=adc^[show,xw]

```



```

        else yw:=tdc^[show,xw];
        DrawPoint(xw,yw);
    end;
    SwapScreen;
    SelectScreen(1);
    RestoreWindow(Inlay,0,0);
    SelectWorld(4);
    SelectWindow(Inlay);
    DrawTextW(3,12,1,'Nr. of ADC/TDC:'); DrawTextW(21,12,1,convert(show,2));
    DrawTextW(3,10,1,'cycles:'); DrawTextW(11,10,1,convert(cycles,3));
    DrawTextW(3,8,1,'scale:'); DrawTextW(10,8,1,convert(scal,7));
    DrawTextW(3,2,1,'mode:'); if modeadc then DrawTextW(10,2,1,'ADC')
                                else DrawTextW(10,2,1,'TDC');
    DrawTextW(18,4,1,'|'); DrawTextW(20,4,1,convert(che,4));
    DrawTextW(3,4,1,'channels:'); DrawTextW(13,4,1,convert(cha,4));
    DrawTextW(3,6,1,'maximum:'); DrawTextW(12,6,1,convert(maxi,5));
    DrawTextW(27,6,1,'('); DrawTextW(28,6,1,convert(mac,4));
    DrawTextW(32,6,1,')');
    REPEAT
    wort:=getin;
    if wort in ['p','P'] then HardCopy(false,1);
    until wort in ['q','Q'];
    end;
:
procedure Plot;
{-----}
var
    i,j,xw,yw:integer;
    madc,mtdc,ascal,tscal,mac,mtc,afac,tfac:integer;
begin
    SelectScreen(2);
    ClearScreen;
    maxi:=0; scal:=hoehe; scflag:=ADCs;
    for i:=adca to adce do
        if (adc^[show,i]>maxi) and (i<1919) then begin
            maxi:=adc^[show,i];
            mac:=i;
        end;
    if not fixed then
        while (scal<maxi) and (scal<16384) do scal:=scal*2;
    madc:=maxi; ascal:=scal; afac:=1;
    if maxi>16384 then begin
        ascal:=32767;
        afac:=2;
    end;
    RestoreWindow(ADCs,0,0);
    DefineWorld(ADCs,adca,0,adce,scal);
    SelectWorld(ADCs);
    SelectWindow(ADCs);
    for xw:=adca to adce do begin
        yw:=trunc(adc^[show,xw]/afac);
        if yw<scal then DrawPoint(xw,yw);
    end;
    maxi:=0; scal:=64; scflag:=TDCs;
    for i:=tdca to tdce-1 do
        if (tdc^[show,i]>maxi) and (i<1919) then begin
            maxi:=tdc^[show,i];
            mtc:=i;
        end;
    while (scal<maxi) and (scal<16384) do scal:=scal*2;
    mtdc:=maxi; tscal:=scal; tfac:=1;
    if maxi>16384 then begin

```

```

tscal:=32767;
tfac:=2;
end;
RestoreWindow(TDCs,0,0);
DefineWorld(TDCs,tdca,0,tdce,scal);
SelectWorld(TDCs);
SelectWindow(TDCs);
for xw:=tdca to tdce do begin
  yw:=trunc(tdc^[show,xw]/tfac);
  if yw<scal then DrawPoint(xw,yw);
end;
RestoreWindow(Info,0,0);
SelectWindow(Info);
wr(64,11,' c : cont. ');
wr(64,12,' q : quit ');
wr(26,4,'          ADC / TDC   Nr:'); wr(55,4,convert(show,2));
wr(11,6,'Aufgenommene ADC/TDC:'); wr(37,6,convert(ach+1,4)); wr(43,6,'/');
wr(46,6,convert(tch+1,4));
wr(11,7,'Skalierung der Counts:'); wr(37,7,convert(ascal,5)); wr(44,7,'/');
wr(47,7,convert(tscal,5));
wr(11,8,'maximaler Inhalt:'); wr(37,8,convert(madc,5)); wr(44,8,'/');
wr(47,8,convert(mtdc,5));
wr(11,9,'Kanalnr. dieses Maximums:'); wr(37,9,convert(mac,4)); wr(43,9,'/');
wr(46,9,convert(mtc,4));
wr(11,10,'Kanaele von|bis:'); wr(32,10,convert(adca,4)); wr(36,10,'|');
wr(37,10,convert(adce,4)); wr(43,10,'/');
wr(46,10,convert(tdca,4)); wr(50,10,'|'); wr(51,10,convert(tdce,4));
wr(11,11,'Anzahl der Messdurchlaufe ('); wr(39,11,convert(cycle,5));
wr(45,11,'Events):');
wr(58,11,convert(cycles,3));
SwapScreen;
SelectScreen(1);
end;

```

Die Routinen in dem File Select.inc realisieren die in den Menüs angebotenen Optionen. Das Hauptmenü und das Menü in dem man die Form der Spektren verändern kann sind in Abbildung 5.3 dargestellt.

MENU		
WatE 5.11/88		
Filename:	F	single Big Plot: B
Parameters:	P	
Collect:	C	
Store Spectra:	S	
Restore:	R	
Order:	O	
Exit:	E	

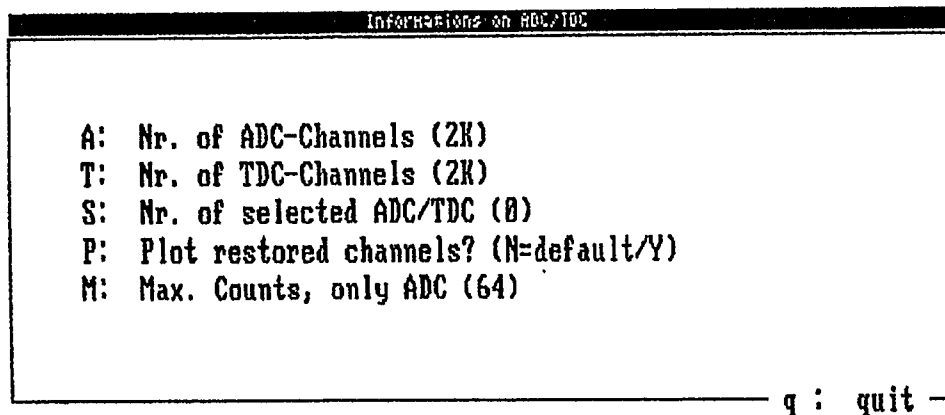


Abbildung 5.3: (a) Hauptmenü, (b) das zu 'P'arameters gehörige Menü mit welchen die Spektren beeinflußt werden.

(• Select.INC)

```

:
procedure Set_Parameters;
var i:integer;
    a:char;
begin
  ClearScreen;
  RestoreWindow(Info,0,0);
  SelectWindow(Info);
  wr(62,12,' q : quit ');
  wr(15,5,'A:  Nr. of ADC-Channels (2K)');
  wr(15,6,'T:  Nr. of TDC-Channels (2K)');
  wr(15,7,'S:  Nr. of selected ADC/TDC (0)');
  wr(15,8,'P:  Plot restored channels? (N=default/Y)');
  wr(15,9,'M:  Max. Counts, only ADC (64)');
:
  REPEAT
    wort:=getin;
    case wort of
      'a','A': begin
        wr(15,11,'from: ---- to: ----');
        i:=getarg(21,11,4);
        if (i>0)and(i<=depth) then adca:=i;
        i:=getarg(30,11,4);
        if (i>0)and(i<=depth) then adce:=i;
        wr(15,11,'
        wort:=' ');
        end;
      't','T': begin
        wr(15,11,'from: ---- to: ----');
        i:=getarg(21,11,4);
        if (i>0)and(i<=depth) then tdca:=i;
        i:=getarg(30,11,4);
        if (i>0)and(i<=depth) then tdce:=i;
        wr(15,11,'
        wort:=' ');
        end;
      's','S': begin

```

```

        wr(15,11,'Nr.: --');
        i:=getarg(20,11,2);
        if (i>-1)and(i<=ach) then show:=i;
        wr(15,11,'      ');
        wort:=' ';
        end;
    'm','M': begin
        wr(15,11,'max.: ----');
        i:=getarg(21,11,4);
        if i>0 then begin
            hoehe:=i;
            fixed:=true;
            end
            else begin
                hoehe:=64;
                fixed:=false;
                end;
            wort:=' ';
        wr(15,11,'      ');
        end;
    'p','P': begin
        plot;
        repeat
            wort:=getin;
            if wort in ['p','P'] then HardCopy(false,1);
            until wort in ['q','Q'];
            wort:=' ';
        end;
    end;
    UNTIL wort in ['q','Q'];
end;
:
procedure Get_Filename;
var f:names;
begin
    {continue:=false;}
    ClearScreen;
    RestoreWindow(Info,0,0);
    SelectWindow(Info);
    wr(60,10,'q: quit');
    wr(20,7,'Please enter filename:');
    wr(30,9,'-----'); f:=getstring(30,9,12);
    if f<>' ' then filename:=f;
    wr(20,7,'Please enter text of information:');
    wr(15,9,'-----');
    txt:=getstring(15,9,50);
    if txt='' then txt:='no information specified !';
end;
:
procedure Store_Spectra;
var i,j:integer;
    suma,sumt:array[0..ach] of real;
    name:names;
begin
    name:=filename + '.DAT';
    Assign(filt,name);
    ReWrite(filt);
    writeln(filt,'number of ADCs: ',ach);
    writeln(filt,'number of TDCs: ',tch);
    writeln(filt,'number of channels: ',depth);
    writeln(filt,'number of cycles: ',cycles);
    writeln(filt,'text:',txt);

```

```

for i:=1 to ach do begin
  for j:=0 to depth do write(filt,adc^[i,j], ' ');
  writeln(filt, ' ');
end;
for i:=1 to tch do begin
  for j:=0 to depth do write(filt,tdc^[i,j], ' ');
  writeln(filt, ' ');
end;
for i:=0 to ach do suma[i]:=0;
for i:=0 to tch do sumt[i]:=0;
writeln(filt, '<---');
for i:=1 to depth do begin
  for j:=0 to ach do suma[j]:=suma[j]+adc^[j,i];
  for j:=0 to tch do sumt[j]:=sumt[j]+tdc^[j,i];
end;
for i:=0 to ach do writeln(filt, '::::', suma[i], 'over:', aoverflow[i]);
for i:=0 to tch do writeln(filt, '-- --', sumt[i], 'over:', toverflow[i]);
Close(filt);
end;
:
procedure Restore_Spectra;
var a1:string[16];a5:string[18];a2:string[20];a3:string[5];a4:names;
    a6:string[24];a:char;
    i,j:integer;
    name:names;
begin
  Get_Filename;
  name:=filename + '.DAT';
  Assign(filt,name);
  Reset(filt);
  readln(filt,a1,i);
  readln(filt,a1,i);
  readln(filt,a2,i);
  readln(filt,a5,cycles);
  readln(filt,a3,a4);
  for i:=1 to ach do begin
    for j:=1 to depth do read(filt,adc^[i,j]);
    readln(filt,a);
  end;
  for i:=1 to tch do begin
    for j:=1 to depth do read(filt,tdc^[i,j]);
    readln(filt,a);
  end;
  Close(filt);
end;
:
procedure Clear;
begin
  continue:=false;
end;
:

```

Das folgende File enthält alle direkt auf das CAMAC-System zugreifenden Routinen. Die eigentliche Meßwertaufnahme erfolgt in der Prozedur GetArray.

(• Collect.INC)

:

```

const
    eps= 256; ece= 512; een=1024;
    cps=2048; cce=4096; csr=8192; cle=16384; ofs=32767;    {ofs +1 !}
                                     {bits of statusregister}
    SetStatusRegister=16; ReadStatusRegister=0; Test=25;
    ReadOut=2; LoadPedestal=17; ReadPedestal=1; ClearCamac=9;
    TestLam=8; TestClearLam=10; enable=26; disable=24;
                                     {commands}
    InitCrate=64; ExecuteZ=1; ExecuteC=2; All=115;
                                     {cycle-commands}

var
    timing=10000;
    crate,station,command,address:integer;
    data,status_q,status_x:integer;
    lam,f:integer;
{-----}
procedure Init_Arrays;
var i,j:integer;
begin
    for i:=1 to depth do begin
        for j:=0 to ach do   adc^[j,i]:=0;
        for j:=0 to tch do   tdc^[j,i]:=0;
        end;
        for i:=0 to param do begin
            aoverflow[i]:=0;
            toverflow[i]:=0;
            end;
        for i:=1 to cycle do
            for j:=1 to param do event^[i,j]:=0;
            end;
end;

procedure CAMACGET;
begin
    CAMI(station,command,address,data,status_q,status_x);
end;

procedure CAMACPUT;
begin
    CAMO(station,command,address,data,status_q,status_x);
end;

function PutPEDESTAL(i:integer):integer;
var j,f:integer;
begin
    f:=ExecuteC;
    CAMCL(f);
    command:=LoadPedestal;
    address:=i;
    data:=0;
    j:=0;
    repeat
        CAMACPUT;
        j:=j+1*round(1/1);
    until (status_q=1) or (j=4000);
    if j=4000 then PutPEDESTAL:=-1
    else PutPEDESTAL:=1;
    CAMCL(f);
end;

function GetPEDESTAL(i:integer):integer;
var j,f:integer;
begin
    f:=ExecuteC;
    CAMCL(f);
    command:=ReadPedestal;

```

```

address:=i;
j:=0;
REPEAT
  CAMACGET;
  j:=j+1*round(1/1);
  UNTIL (status_q=1)or(j=4000);
if status_q=1 then GetPedestal:=data
  else GetPedestal:=-1;
CAMCL(f);
end;
;
function WfL:boolean;
var time,t:integer;
begin
  time:=0;
  status_q:=0;
  address:=0;
  command:=TestLam;
  repeat
    time:=time+1;
    CAMACGET;
    until (status_q=1) or (time>timing);
  if (time<timing) then begin
    WfL:=true; end
  else WfL:=false;
  end;
;
procedure Init_crate;
var i:integer;
    filp:text;
begin
  crate:=1;
                                CRATE_SET(crate);
  f:=ExecuteZ; CAMCL(f);
  f:=InitCrate; CAMCL(f);
  f:=ExecuteC; CAMCL(f);
  address:=0;
  station:=ast;
  data:= cle + station;
  command:=SetStatusRegister;
                                CAMACPUT;
  station:=tst;
  command:=enable;
                                CAMACPUT;
                                CAMCL(f);
  {assign(filp,'pedestal.inf');
  rewrite(filp);
  station:=ast;
  for i:=0 to ach do writeln(filp,i:3,'-',putPEDESTAL(i):4);
  for i:=0 to ach do writeln(filp,i:3,':',getPEDESTAL(i):4);
                                CAMCL(f);
  close(filp);}
  end;
;
procedure Collect_Data;
{-----}
var
  nr,ver, ix,jx:integer;
;
procedure GetArray;
var i,j,f:integer;
begin
  f:=ExecuteC; CAMCL(f);

                                { ---> read ADC's }

  station:=ast;
  repeat until WfL;

```

```

command:=ReadOut;
for i:=0 to ach do begin
  address:=i;
  CAMACGET;
  if (data>0) and (data<1919) then begin
    event^[nr,i+1]:=data;
    if adc^[i,data]<32000 then adc^[i,data]:=adc^[i,data]+1;
  end
  else aoverflow[i]:=aoverflow[i]+1;
end;
{ ---> read TDC's }
station:=tst;
repeat until WfL;
command:=ReadOut;
for i:=0 to tch do begin
  address:=i;
  CAMACGET;
  if (data>0) and (data<1919) then begin
    event^[nr,i+2+ach]:=data;
    if tdc^[i,data]<32000 then tdc^[i,data]:=tdc^[i,data]+1;
  end
  else toverflow[i]:=toverflow[i]+1;
end;
f:=ExecuteC; CAMCL(f);
end;
::
begin
  if not continue then begin
    filename:=filename + '.INF';
    Assign(filt,filename);
    Rewrite(filt);
    writeln(filt,' cycle: ',cycle,' parameter: ',param);
    writeln(filt,' ADCs: ',ach,' TDCs: ',tch,' Channels ',depth);
    filename:=filename + '.EVE';
    Assign(filvar,filename);
    Rewrite(filvar);
    cycles:=0;
    Init_Arrays;
  end
  else plot;
REPEAT
  cycles:=cycles+1;
  for nr:=1 to cycle do GetArray;
  Write(filvar,event^);
  plot;
  UNTIL (cycles>=freq) or keypressed;
REPEAT
  if cycles<freq then wort:=getin
  else wort:='Q';
  case wort of
    'c','C': continue:=true;
    'p','P': HardCopy(false,1);
    'q','Q': begin
      continue:=false;
      writeln(filt,'cycles: ',cycles);
      Close(filt);
      Close(filvar);
      if wort='Q' then Store_Spectra;
    end;
  end;
  UNTIL wort in ['q','Q','c','C'];
  wort:=' ';
end;

```


Abschließend soll hier eine Hardcopy des 'online' Bildschirms (Abbildung 5.4) wiedergegeben werden. Man kann jeweils ein gewähltes Zeit- und Amplitudenspektrum während der Messung beobachten. Die Form in der einzelne Spektren graphisch festgehalten werden können zeigt Figur 5.5.

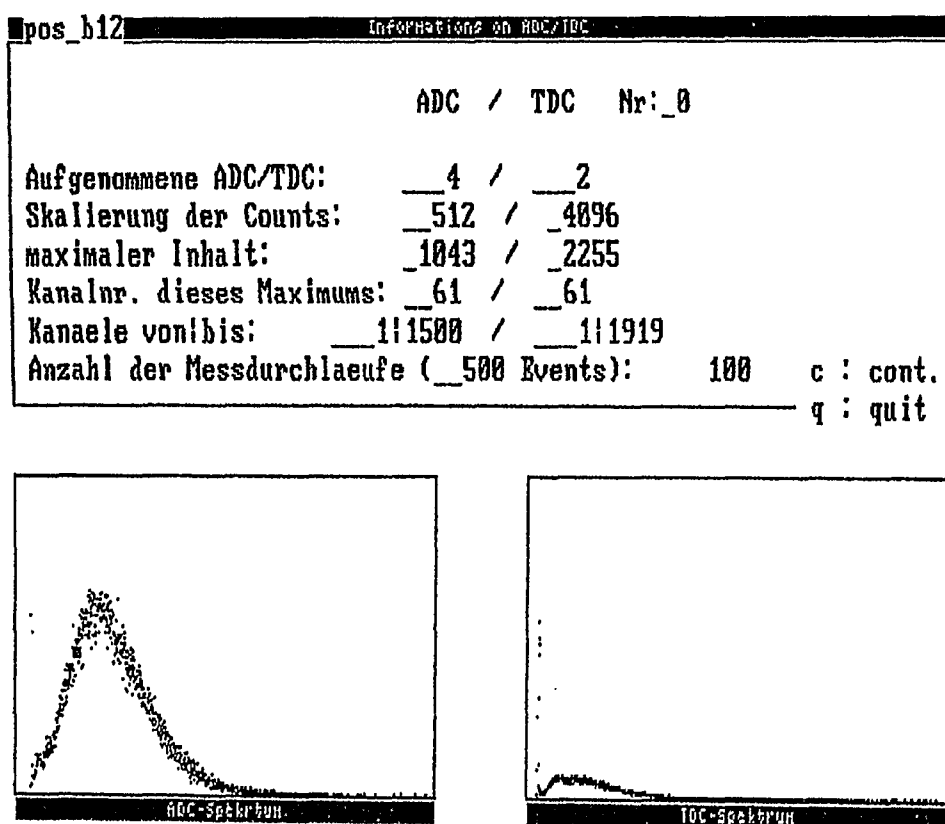


Abbildung 5.4: Gestalt des Bildschirms während einer Messung.

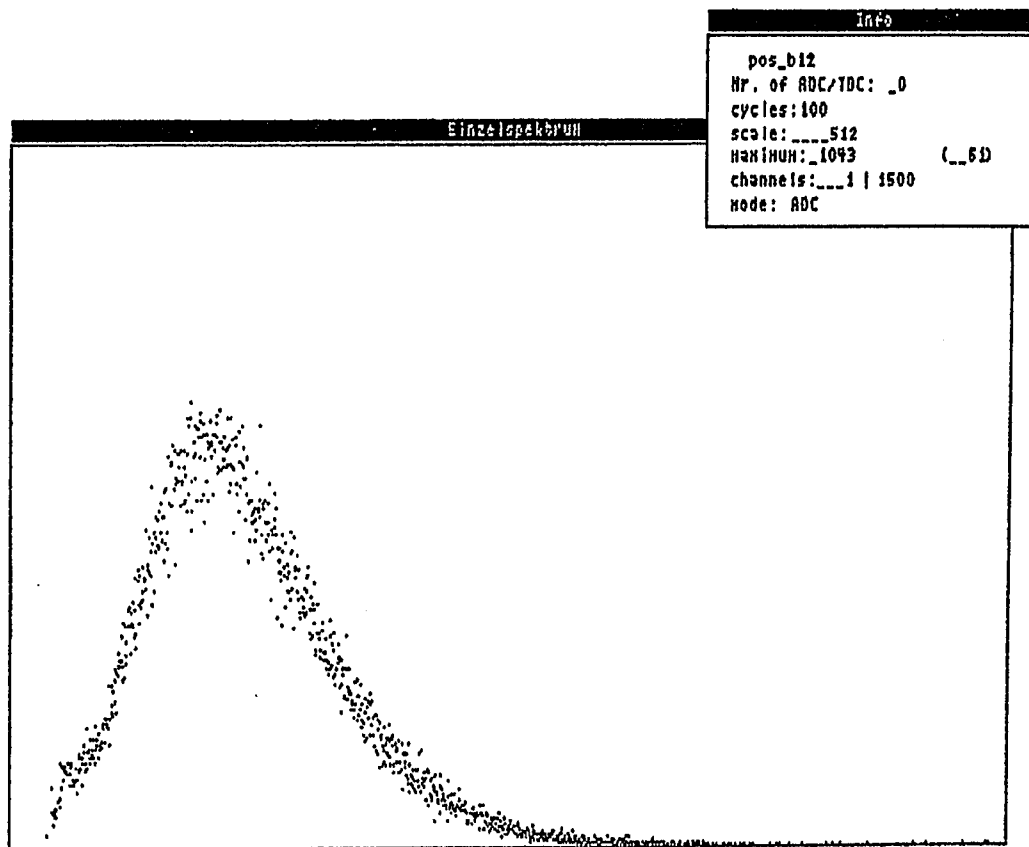


Abbildung 5.5: Einzelspektrum, mit der Möglichkeit durch erneutes betätigen der Taste P eine Hardcopy zu erzeugen.

C Ergänzung zu Kapitel 2

- (i) Wahrscheinlichkeits-Transformation
- (ii) Seitennumerierung für einen Quader
- (iii) opaq exec
- (iv) opaquef fortran
- (v) Datenfile für den simulierten Versuchsaufbau

(i) Wahrscheinlichkeits-Transformation

Eine gleichverteilte Zufallszahl ist durch

$$p(x)dx = \begin{cases} dx & \text{falls } 0 \leq x \leq 1 \\ 0 & \text{sonst} \end{cases} \quad (1)$$

gegeben. Die Wahrscheinlichkeit eine Zahl zwischen x und $x + dx$ zu finden, wird dabei mit $p(x)dx$ bezeichnet. Es gilt natürlich die Normierung der Wahrscheinlichkeit auf 1:

$$\int_{-\infty}^{\infty} p(x) dx = 1 \quad (2)$$

Erzeugt man jetzt eine solche gleichverteilte Zufallszahl x und wendet auf sie eine Funktionsvorschrift $y(x)$ an, so ist die Wahrscheinlichkeitsverteilung von y , mit $p(y)dy$ bezeichnet, durch das fundamentale Transformationsgesetz für Wahrscheinlichkeiten gegeben:

$$\begin{aligned} |p(x)dx| &= |p(y)dy| \\ p(y) &= p(x) \left| \frac{dx}{dy} \right| \end{aligned} \quad (3)$$

Benutzt man jetzt die Voraussetzung (1), daß x eine gleichverteilte Zufallszahl ist, so findet man:

$$\begin{aligned} |p(y)dy| &= |p(x)dx| \stackrel{!}{=} |dx| = \left| \frac{dx}{dy} \right| dy \\ \Rightarrow p(y) &= \left| \frac{dx}{dy} \right| \end{aligned} \quad (4)$$

Möchte man nun eine bestimmte Wahrscheinlichkeitsverteilung $f(y)$ (f sei auf dem betrachteten Intervall invertierbar und normierbar) aus der gleichverteilten Zufallszahl x erhalten, so folgt für die Transformation:

$$f(y) = p(y) = \left| \frac{dx}{dy} \right| \Rightarrow f(y) = \left| \frac{dx}{dy} \right| \quad (5)$$

Die Lösung dieser Differentialgleichung lautet:

$$x = F(y) \quad \text{mit} \quad F(x) := \int_0^x f(y) dy$$

und die gesuchte Transformation ist damit:

$$y(x) = F^{-1}(x) \quad (6)$$

Das bedeutet, um eine Wahrscheinlichkeitsverteilung $f(y) = p(y)$ zu erhalten, muß man auf die gleichverteilte Zufallszahl x die Transformation $y(x) = F^{-1}(x)$ anwenden.

Gesucht war eine sinusförmige Wahrscheinlichkeitsverteilung. Für diese folgt mit

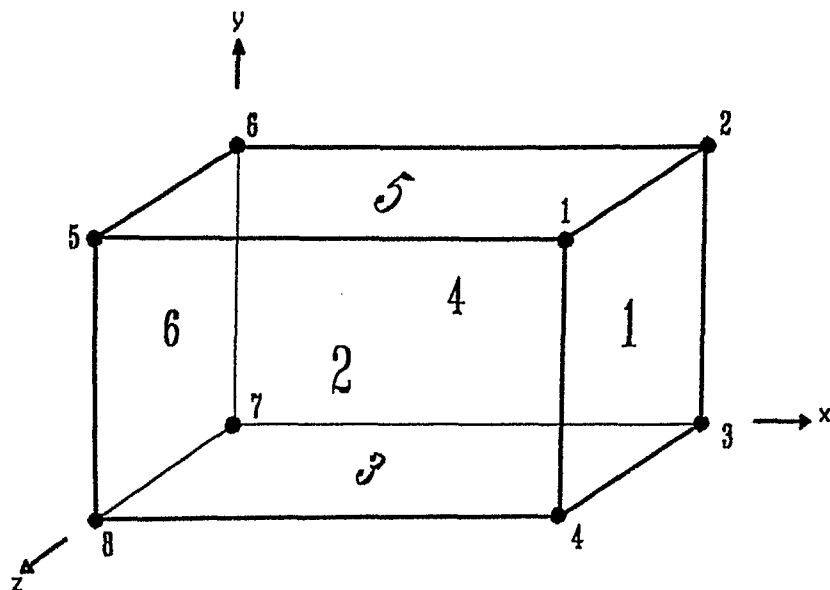
$$f(y) = p(y) = \sin(y) \quad [0 \leq y \leq \pi]$$

und Transformation (6):

$$F^{-1} = (\cos(y))^{-1} = \arccos(x)$$

[PVTf87]

(ii)



Für den einfachen Fall eines rechteckigen Körpers zeigt obige Abbildung die im Programm gewählte Bezeichnung der Eckpunkte. Sie werden in der Reihenfolge $1 \rightarrow 8$ eingegeben. Die Zuordnung der jeweiligen Nummer zu den Seitenflächen

(deren Lage im Koordinatensystem nur relativ festgelegt ist) wird über die Eingabe der Eckkoordinaten vom Benutzer festgelegt.

(iii)

Das in REXX geschriebene Programm OPAQ dient zum Ausführen von OPAQUE. Es ruft nach den nötigen Filedefinitionen für die Eindabedateien und dem Erstellen einer Datei GRDEVICE GRDEVICE (sie wird von System zur Workstation-Initialisierung benötigt) das Fortran Program OPAQUEF auf.

(• OPAQ EXEC)

```

;
/*.....
/.....
/.....      Diese EXEC startet das
/.....      Programmpaket
/.....
/.....      OPAQUE
/.....
/.....*/
arg name
if name='' then parse upper value name with fn ft fm
else do
    say 'default Datensatz gelesen'
    fn='OPAQ'
    ft='DATA'
    fm='A'
end
say , ----- ,
say 'inputfile DATA:      'fn ft fm
say , ----- ,
'FILEDEF FIL DISK' fn ft fm
/*..... lesen fn ft fm ...DATA...*/
'execio 1 disk' fn ft fm '2'
parse pull name
parse upper value name with fn1 ft1 fm1 rest
say , ----- ,
say 'inputfile SPEKTREN:      'fn1 ft1 fm1
say , ----- ,
'FILEDEF SPE DISK' fn1 ft1 fm1
/*..... lesen fn ft fm ...SPEC...*/
'execio 1 disk' fn ft fm '3'
parse pull name
parse upper value name with fn1 ft1 fm1 rest
say , ----- ,
say 'outputfile ERGEBNIS:      'fn1 ft1 fm1
say , ----- ,
'FILEDEF ERG DISK' fn1 ft1 fm1 '(LRECL 1256 RECFM V'
/*..... lesen fn ft fm ...ERGE...*/
'execio 1 disk' fn ft fm '4'
parse pull a
parse value a with b rest
say b
/*.....*/
'cp set emsg off'
'state GRDEVICE GRDEVICE A'
if rc=0 then 'erase GRDEVICE GRDEVICE A'
'cp set emsg on'

```


D Simulationsprogramm OPAQUE

Dem Anwender stehen in der Routine *sort* alle Variablen des Hauptprogramms zur Verfügung. Die Punkte am linken Rand gehören nicht zum Programmtext.

```

:
segment opaque;
  procedure opaque; main;      {---- 25.04.1989 ----}
  procedure opaque;
{<<-----<<<<----->>>>----->>}
:
const
  x=1; y=2; z=3;                                {|koordinaten|}
  activ=1; passiv=0; finit=2;                    {|zustand der flaechen|}
  zero=1E-7;                                     {|un-genauigkeit des rechners|}
  null=1E-12;                                    {|entspricht null|}
  light=29.97924;                               {|lichtgeschwindigkeit (cm/nsec)|}
  attenuation=125;                              {|daempfungslaenge im sz.|}
  nluft=1; nmedium=1.5;                         {|brechungsindizes|}
  quader=1; spiral=2;                           {|shaping|}
  shapes=3;
  bodies=20;

type
  vector= array[1..3] of real;
  pl= array[1..3] of vector;
  identity= array[1..bodies,1..6,0..4] of vector;

var
  i,histogramme : integer;
  titel : array[1..4] of string(30);
  xmin,xmax,ymin,ymax : array[1..4] of real;
  xbins,ybins : array[1..4] of integer;
  xdelta,ydelta : array[1..4] of real;
  xbounds : array[1..4,0..205] of real;
  ybounds : array[1..4,0..205] of real;
  counts : array[1..4,1..205,1..205] of real;      {| ---> order|}
  koor,number:integer;
  NrofPhotons,photon,body,MrBody,nr,snr,PROPERTY,Reflections,
  nix,SubBody,side,location : integer;
  t,alpha,PROBABILITY,attenuate,way,time,loss : real;
  old_p,origin,refl,point,aim,polaris : vector;
  decline,REFLECTED,total : boolean;
  Grenzwahrscheinlichkeit,SurfaceLoss : real;
  shape : array[1..bodies] of integer;
  rang : array[1..shapes,1..bodies] of integer;
  index : array[1..shapes,1..bodies] of real;
  interf : array[1..shapes,1..bodies,1..6] of integer;
  arg,fac,pi,xx,yy, werte,times : real;           {|globale!!}
  tim,date : alfa;
  tube,murks : integer;
  id : identity;
  outside,interactive : boolean;
  lastbody:integer;
  plane : pl;

```



```

mode : array[1..bodies,1..6] of integer;
betrag_id : array[1..bodies,1..6,0..4] of real;
plain : array[1..bodies,1..6] of real;
ma,mi : vector;                                { | --->quader | }
rho_0,rho_d,seg,rad_0,zmin,zmax, mrad : real;
head : pl;
stirn : array[0..3] of vector;
mode_stirn : integer;
posx,posy,dx,dy,MINx : real;                    { | --->spiral | }
paint:integer;
outp,ddd : text; points : array[0..20000,1..2] of real;
w1,w2,w3 : integer;
cphi,sphi,crho,srho,cxsi,sxsi : real;           { | --->graphic | }
scale,asin:real;
{>>-----<<>----- mathe txtlib -----<< }
function square(var a:vector):real;
  fortran;
function determ(var a,b,c:vector):real;
  fortran;
function amount(var a:vector):real;
  fortran;
function scalar(var a,b:vector):real;
  fortran;
function factor(var a,b:vector; var plane:pl):real;
  fortran;
procedure ortho(var a,b,c:vector);
  fortran;
function cos(x:real):real;
function cosf(var x:real):real;
  fortran;
begin
  cos:=cosf(x);
end;
function sin(x:real):real;
function sinf(var x:real):real;
  fortran;
begin
  sin:=sinf(x);
end;
function arccos(var x:real):real;
  fortran;
function arcsin(var x:real):real;
  fortran;
{===== graphic =====}
var colour:integer;
procedure grsclc(var x1,y1,x2,y2:real);
  fortran;
procedure grsclv(var x1,y1,x2,y2:real);
  fortran;
procedure grjmp(var x,y:real);
  fortran;
procedure grdrw(var x,y:real);
  fortran;
procedure grnwpn(var x:integer);
  fortran;
procedure grspts(var x:integer);
  fortran;
procedure grnxtf;
  fortran;

```

```

procedure view;
  var phi,rho,xsi,lx,ly,rx,ry:real;
begin
  phi:= w1 /180*pi;
  rho:= w2 /180*pi;
  xsi:= w3 /180*pi;
  cphi:=cos(phi);sphi:=sin(phi);
  crho:=cos(rho);srho:=sin(rho);
  cxsi:=cos(xsi);sxsi:=sin(xsi);
  lx:=-80; rx:=80; ly:=-60; ry:=60;
  grsclv(lx,ly,rx,ry);
end;

function fktz(a:vector):real;
begin
  fktz:=-a[x]*srho + a[y]*crho*sphi - a[z]*crho*cphi;
end;

function fktx(a:vector):real;
begin
  fktx:= a[x]*cxsi*crho + a[y]*(-sxsi*cphi+cxsi*srho*sphi)
        + a[z]*( sxsi*sphi+cxsi*srho*cphi);
end;

function fkty(a:vector):real;
begin
  fkty:= a[x]*sxsi*crho + a[y]*( cxsi*cphi+sxsi*srho*sphi)
        + a[z]*(-cxsi*sphi+sxsi*srho*cphi);
end;

procedure move;
var i:integer;
    xw1,yw1:real;
begin
  i:=30; grspts(i);
  colour:=6; grnwprn(colour);
  grjmp(points[0,x],points[0,y]);
  grdrw(points[0,x],points[0,y]);
  grjmp(points[0,x],points[0,y]);          {/// GR murks ///}
  colour:=3; grnwprn(colour);
  for i:=1 to Reflections do grdrw(points[i,x],points[i,y]);
  i:=16; grspts(i);
end;

{===== graphic =====}
{===== order =====}

procedure order_init;
var k,i,j:integer;
begin
  for k:=1 to histogramme do begin
    for i:=1 to xbins[k] do
      for j:=1 to ybins[k] do counts[k,i,j]:=0;
    xdelta[k]:=(xmax[k]-xmin[k])/xbins[k];
    ydelta[k]:=(ymax[k]-ymin[k])/ybins[k];
    for i:=0 to xbins[k] do xbounds[k,i]:=xmin[k]+ i *xdelta[k];
    for i:=0 to ybins[k] do ybounds[k,i]:=ymin[k]+ i *ydelta[k];
  end;
end;

procedure order(k:integer;x,y,z:real);
var xindex,yindex:integer;
begin
  if (x>=xmin[k])and(x<=xmax[k])and

```



```

e2[x]:= -aim[x] * aim[z];   e2[y]:= -aim[y] * aim[z];
e2[z]:=sqr(aim[x])+sqr(aim[y]);
e1b:=amount(e1); e2b:=amount(e2);
for koor:=x to z do begin
  e1[koor]:=e1[koor]/e1b;
  e2[koor]:=e2[koor]/e2b;
end;
f1:=rndm(nix); f2:=rndm(nix);
for koor:=x to z do polaris[koor]:=f1*e1[koor]+f2*e2[koor];
pb:=amount(polaris);
for koor:=x to z do polaris[koor]:=polaris[koor]/pb;
end;
{.....Polarisation.....}
function interface(shape,SubBody,side:integer):integer;
{////////////////////////////////////}
begin
  interactive:=true;
  lastbody:=body;
  interface:=interf[shape,SubBody,side];
end;
{.....InterFace.....}
procedure Plexi(SubBody,side:integer);
{////////////////////////////////////}
const  bodies= 11;
var    b:integer;
begin
  b:=0;
  IF (body=2)and(side in [2,4,6]) THEN begin
    WHILE (b<=bodies) and not (( refl[y]<id[b+1,1,0,y])
                                and( refl[y]>id[b+1,1,1,y])) DO b:=b+1;
    IF b>bodies THEN begin
      mode[SubBody,side]:=passiv;
{-----
      PROBABILITY:=PROBABILITY*0.50;
-----}
      end
      ELSE begin
        tube:=b;
        mode[SubBody,side]:=activ;
        interf[quader,SubBody,side]:=b+2;
{-----
        PROBABILITY:=PROBABILITY*0.85;
-----}
        end;
      end;
    end;
  end;
{.....Plexi.....}
function Attenuated(P:real):real;
{////////////////////////////////////}
var delta:vector;
    i:integer;
begin
  for i:=x to z do begin
    delta[i]:=point[i]-old_p[i];
    old_p[i]:=point[i];
  end;
  loss:=loss*Surfaceloss;
  way:=way+amount(delta);
  Attenuated:=P*exp(-way/attenuation)*loss;

```

```

end;
{.....Attenuate.....}
procedure sort;
{////////////////////////////////}
var a,ay,az:real;
begin
  { a :=amount(aim);
    ay:=aim[y]/a; az:=aim[z]/a;
    order(1,az,ay,attenuate);
    order(2,origin[x],origin[y],attenuate);
    order(3,point[y],point[z],attenuate); }
    order(1,0,tube,attenuate);
    order(2,tube,attenuate,1);
end;
{.....FILL.spectra.....}
;
procedure dump(a:vector);
{////////////////////////////////}
begin
  writeln(a[x]:3:6,' ',a[y]:3:6,' ',a[z]:3:6);
end;
{===== quader =====}
;
procedure graphic_quader(body:integer);
{////////////////////////////////}
var s,j,c : integer;
begin
  c:=2; grnwpn(c);
  for s:=6 downto 1 do begin
    xx:=fktz(id[body,s,0]); yy:=fktz(id[body,s,0]);
    grjmp(xx,yy);
    for j:=1 to 4 do begin
      xx:=fktz(id[body,s,j]); yy:=fktz(id[body,s,j]);
      grdrw(xx,yy);
    end;
  end;
  c:=3;
  grnwpn(c);
end;
{.....}
function area(pkt:vector;body,surface:integer) : real;
{////////////////////////////////}
var height,areal,sc,a,b,c:real;
    edge,abst:array[0..4] of real;
    n,koor:integer;
    dist:vector;
    {////////////////////////////////}
    {/ berechnet Mittels cos-satz die abstaende /}
    {/ des punktes von den ecken. /}
    {/ dann werden die flaechen der aus jeweils /}
    {/ zwei ecken und pkt bestehenden Zecke /}
    {/ berechnet und zur gesamtflaeche addiert. /}
    {/ ----> input , point_quader /}
    {////////////////////////////////}
begin
  areal:=0;
  for n:=0 to 3 do
    begin

```

```

    sc:=2*scalar(id[body,surface,n],pkt);
    abst[n]:=betrag_id[body,surface,n]+square(pkt)-sc;
    sc:=2*scalar(id[body,surface,n],id[body,surface,n+1]);
    edge[n]:=square(id[body,surface,n])+square(id[body,surface,n+1])-sc;
    end;
    abst[4]:=abst[0];
  for n:=0 to 3 do
    begin
      a:=abst[n]; b:=abst[n+1]; c:=edge[n];
      height:=sqrt(abs( a-sqr(a-b+c)/(4*c) ));
      areal:=areal+height*sqrt(c)/2;
    end;
  area:=areal;
end;
{.....}
procedure point_quader(bodi:integer);
  forward;
{.....}
procedure Generate_Quader;
{\\////////////////////////////////////}
  var koor,i,j:integer;
      {////////////////////////////////////}
      {/ es wird auf in einem 'ueberquader' ein /}
      {/ punkt gewuerfelt mit zugehoeriger rich- /}
      {/ tung. /}
      {/ findet man 2 (!) reflexionspunkte auf dem /}
      {/ echten koerper, so muss der generierte /}
      {/ punkt innerhalb liegen. /}
      {/ ----> main /}
      {////////////////////////////////////}
begin
  REPEAT
    for koor:=x to z do
      point[koor]:=rndm(nix)*(ma[koor]-mi[koor])+mi[koor];
    richtung(aim);
    point_quader(1);
    if not outside then begin
      for koor:=x to z do aim[koor]:=-aim[koor];
      point_quader(1);
      for koor:=x to z do aim[koor]:=-aim[koor];
      end;
    UNTIL not outside;
  end;
{.....}
procedure point_quader;      {(bodi:integer);}
{\\////////////////////////////////////}
  var koor,surf : integer;
      {////////////////////////////////////}
      {/ diese routine findet den reflexions- /}
      {/ punkt und die dazugehoerige -ebene. /}
      {/ mittels determinantenmethode geloestes /}
      {/ g/e schnittpunkts-gleichungssystem. /}
      {/ factor: abstand zur ebene /}
      {/ 'area-' flaechenvergleich (!) /}
      {/ ----> input , reflect /}
      {////////////////////////////////////}
begin
  surf:=0; outside:=false;
  REPEAT

```

```

surf:=surf+1;
if surf>8 then begin
  surf:=surf-1;
  outside:=true;
end;
if interactive and (mode[bodi,surf]=activ)
  and (interf[quader,bodi,surf]=lastbody) then
  if surf<6 then surf:=surf+1
  else outside:=true;
for koor:=x to z do begin
  plane[1,koor]:=id[bodi,surf,2,koor];
  plane[2,koor]:=id[bodi,surf,1,koor]-id[bodi,surf,2,koor];
  plane[3,koor]:=id[bodi,surf,3,koor]-id[bodi,surf,2,koor];
end;
t:=factor(point,aim,plane);
for koor:=x to z do refl[koor]:=point[koor]+t*aim[koor];
UNTIL
  ((abs(plain[bodi,surf]-area(refl,bodi,surf))<=zero
    and (t>null)) or outside;
interactive:=false;
Plexi(bodi,surf);
PROPERTY:=mode[bodi,surf];
side:=surf;
end;
{===== quader =====}

{===== spiral =====}
procedure graphic_spirale;
{\\////////////////////////////////////}
var s,i,j,step,koor:integer;
    vec:vector;
    phi,radius:real;
    zw:array[0..1] of real;
begin
  zw[0]:=zmin;zw[1]:=zmax;
  colour:=2; grwpn(colour);
  for j:=0 to 1 do
  for i:=0 to 1 do begin
    xx:=0; yy:=0; grjmp(xx,yy);
    for step:=0 to 180 do begin
      phi:=(step/180*rho_d+seg*i+rho_0);
      radius:=rad_0*step/180;
      vec[z]:=zw[j]; vec[y]:=radius*sin(phi); vec[x]:=radius*cos(phi);
      xx:=fktx(vec); yy:=fktz(vec);
      grdrw(xx,yy);
    end;
    for koor:=x to z do stirn[j*2+i,koor]:=vec[koor];
  end;
  xx:=fktx(stirn[0]); yy:=fktz(stirn[0]); grjmp(xx,yy);
  xx:=fktx(stirn[1]); yy:=fktz(stirn[1]); grdrw(xx,yy);
  xx:=fktx(stirn[3]); yy:=fktz(stirn[3]); grdrw(xx,yy);
  xx:=fktx(stirn[2]); yy:=fktz(stirn[2]); grdrw(xx,yy);
  xx:=fktx(stirn[0]); yy:=fktz(stirn[0]); grdrw(xx,yy);
end;
{.....}
procedure Generate_Spirale;
{////////////////////////////////////}

```

```

var nix:integer;
    r,phi:real;
    {////////////////////////////////////}
    {/ wuerfelt homogen in einen Kreis, der /}
    {/ auf ein Tortenstueck gefaltet und /}
    {/ dann zu einem 'Quirlsegment' gebogen /}
    {/ wird. /}
    {////////////////////////////////////}
begin
  repeat
    phi:=2*pi*rndm(nix);
    r:=rad_0*sqrt(rndm(nix));
    phi:=phi-trunc(phi/pi*6)*pi/6;
    point[x]:= r*cos(r/rad_0*rho_d+phi+rho_0);
    point[y]:= r*sin(r/rad_0*rho_d+phi+rho_0);
    until (point[x]>MINx);
    point[z]:=zmin+(zmax-zmin)*rndm(nix);
    richtung(aim);
  end;
{.....}
procedure head_spirale;
{////////////////////////////////////}
var i,j,koor:integer;
    vec:vector;
    phi:real;
    zw:array[0..1] of real;
begin
  zw[0]:=zmin;zw[1]:=zmax;
  for j:=0 to 1 do
    for i:=0 to 1 do begin
      phi:=rho_d+seg*i+rho_0;
      vec[z]:=zw[j]; vec[y]:=rad_0*sin(phi); vec[x]:=rad_0*cos(phi);
      for koor:=x to z do stirn[j*2+i,koor]:=vec[koor];
    end;
    for i:=x to z do begin
      head[1,i]:=stirn[0,i];
      head[2,i]:=stirn[1,i]-stirn[0,i];
      head[3,i]:=stirn[2,i]-stirn[0,i];
    end;
  end;
  {stirnflaeche der spirale}
{.....}
procedure point_spirale;
{////////////////////////////////////}
var t,rho : real;
    {////////////////////////////////////}
    {/ die funktion der spirale lautet: /}
    {/ x:=c*rho*cos(rho) y:=c*rho*sin(rho) /}
    {/ leider lassen sich die nullstellen nicht /}
    {/ analytisch berechnen (transzendent!) /}
    {/ --> NEWTON - method /}
    {/ ( gerade-spirale=0 ) /}
    {/ als startpunkte werden die maxima der /}
    {/ ableitung gewaehlt brill! /}
    {////////////////////////////////////}
{-----> start of procedures <-----}
procedure newton(var t,rho:real;seg:real);
{////////////////////////////////////}
var i,iteration:integer;
    maxi,mini,est,ext:real;

```



```

        oxw,nxw:real;
        tw,xw,extrem : array[1..4] of real;
function fkt(rho:real):real;      { | funktion |}
{-----}
begin
    fkt:=rad_0/rho_d*(rho-rho_0-seg)*(aim[y]*cos(rho)-aim[x]*sin(rho))+
        point[y]*aim[x]-point[x]*aim[y];
end;
function dfkt(rho:real):real;      { | ableitung |}
{-----}
begin
    dfkt:=rad_0/rho_d*(aim[y]*cos(rho)-aim[x]*sin(rho)) -
        rad_0/rho_d*(rho-rho_0-seg)*(aim[y]*sin(rho)+aim[x]*cos(rho));
end;
{----- | find entrypoint for newton | -----}
begin
    maxi:=0; mini:=0;
    for i:=0 to 10 do begin
        est:=rho_0+i/10*rho_d+seg;
        ext:=dfkt(est);
        if (maxi < ext) then begin
            maxi:=ext;
            extrem[1]:=est;
        end;
        if (mini > ext) then begin
            mini:=ext;
            extrem[2]:=est;
        end;
        extrem[3]:=rho_0+seg;
        extrem[4]:=rho_0+seg+rho_d;
    end;

    {- | execute newton | -}
    for i:=1 to 4 do begin
        iteration:=0;
        nxw:=extrem[i];
        repeat
            iteration:=iteration+1;
            oxw:=nxw;
            nxw:=(oxw-fkt(oxw)/dfkt(oxw));
            until (abs(oxw-nxw)<null) or (iteration>25);
            extrem[i]:=nxw;
            if iteration>25 then extrem[i]:=9997;
            if (nxw<rho_0+seg) then extrem[i]:=9996;
        end;

        {- | look for t and rho | -}
        tw[1]:=9998; tw[2]:=9998; tw[3]:=9998; tw[4]:=9998; t:=9998;
        for i:=1 to 4 do
            if extrem[i]<9000 then begin
                xw[i]:=rad_0*(extrem[i]-rho_0-seg)/rho_d*cos(extrem[i]);
                if abs(aim[x])>null then tw[i]:=(xw[i]-point[x])/aim[x];
            end;
        for i:=1 to 4 do
            if (tw[i]>null)and(tw[i]<=t) then begin
                t:=tw[i];
                rho:=extrem[i];
            end;
        end;
    {.....}
    procedure Center;
    {//////////}
    var koor:integer;

```

```

    root,a,b,c,t1:real;
begin
  a:=sqr(aim[x])+sqr(aim[y]);
  b:=point[x]*aim[x]+point[y]*aim[y];
  c:=sqr(point[x])+sqr(point[y])-sqr(mrad);
  root:=sqrt(sqr(b)-a*c);
  t:=(-b+root)/a; t1:=(-b-root)/a;
  if (t<null) or ((t1>null)and(t1<t)) then t:=t1;
  for koor:=x to z do begin
    refl[koor]:=point[koor]+t*aim[koor];
    plane[1,koor]:=refl[koor];
  end;
  plane[2,x]:=0; plane[2,y]:=0; plane[2,z]:=1;
  ortho(refl,plane[2],plane[3]);
end;

function Cover(var t:real):boolean;
{////////////////////////////////////}
var koor:integer;
    tz,th, r,xsi,rho,rmax,rmin,max ,a,t1,t2:real;
    accept:boolean;
begin
  Cover:=false;
  if abs(aim[z])<null then aim[z]:=null;
  tz:=(zmax-point[z])/aim[z];
  th:=(zmin-point[z])/aim[z];
  if (th>tz) then tz:=th;
  for koor:=x to z do refl[koor]:=point[koor]+tz*aim[koor];
  accept:=false;
  r:=sqrt(sqr(refl[x])+sqr(refl[y]));
  xsi:=refl[x]/r;
  xsi:=arccos(xsi);
  if refl[y]<0 then xsi:=2*pi-xsi;
  rho:=xsi-rho_0;
  if rho>2*pi then rho:=rho-2*pi;
  rmax:=rad_0*rho/rho_d;
  rmin:=rad_0*(rho-seg)/rho_d;
  if (r<rmax)and(r>rmin)and(rho>0) and
    (point[x]>MIWx) then accept:=true;
  if sqr(aim[z])<((sqr(aim[x])+sqr(aim[y]))/50) then begin
    newton(t1,a,seg);
    newton(t2,a,0);
    if ((t1>null)and(t1<tz))or((t2>null)and(t2<tz)) then accept:=false;
  end;
  if accept then begin
    Cover:=true;
    if sqr(mrad)>sqr(refl[x])+sqr(refl[y]) then Center
  else begin
    t:=tz;
    for koor:=x to z do plane[1,koor]:=point[koor]+t*aim[koor];
    plane[2,x]:=1; plane[2,y]:=0; plane[2,z]:=0;
    plane[3,x]:=0; plane[3,y]:=1; plane[3,z]:=0;
  end;
end;
end;

procedure Sides(var t:real);
{////////////////////////////////////}
var sego,dx,dy, rhoi,rad ,ti: real;
    koor:integer;
begin

```

```

newton(t,rho,seg);
newton(ti,rhoi,0);
sego:=seg;
if ((ti<t)and(ti>null))or(abs(t)<null) then begin
  t:=ti;
  rho:=rhoi;
  sego:=0;
end;
if abs(rho)<null then rho:=rho+null;
dx:=rad_0/rho_d*( cos(rho)-(rho-rho_0-sego)*sin(rho) );
dy:=rad_0/rho_d*( sin(rho)+(rho-rho_0-sego)*cos(rho) );
for koor:=x to z do begin
  refl[koor]:=point[koor]+t*aim[koor];
  plane[1,koor]:=point[koor]+t*aim[koor];
end;
if sqr(mrad)>sqr(refl[x])+sqr(refl[y]) then Center
else begin
  plane[2,x]:=0; plane[2,y]:=0; plane[2,z]:=1;
  plane[3,x]:=dx; plane[3,y]:=dy; plane[3,z]:=0;
end;
end;
procedure QueryHead(t:real);
{//////////}
var facet:real;
    koor:integer;
    inside:boolean;
    vec:vector;
begin
  facet:=9999;
  inside:=false;
  if abs(aim[x])>null then facet:=(MINx-point[x])/aim[x];
  for koor:=x to z do vec[koor]:=point[koor]+facet*aim[koor];
  if (vec[y]>stirn[1,y])and(vec[y]<stirn[0,y])and
    (vec[z]>zmin)and(vec[z]<zmax) then inside:=true;
  if (facet<t) and (facet>null) and inside
    and not interactive then begin
    property:=mode_stirn;
    for koor:=x to z do begin
      refl[koor]:=point[koor]+facet*aim[koor];
      plane[1,koor]:=refl[koor];
      plane[2,koor]:=head[2,koor];
      plane[3,koor]:=head[3,koor];
    end;
  end;
end;
{-----> end of procedures <-----}
begin
  property:=passiv;
  if not Cover(t) then Sides(t);
  QueryHead(t);
  interactive:=false;
end;
{===== spiral =====}
{===== input =====}
procedure input;
{\\}
var fil,spe:text;
    pkt:vector;
    corn:array[1..8,1..3] of real;

```

```

orde:array[1..5,1..6] of integer;
number,qn,cn,i,j,k,koor,interfaces:integer;
F,radius1:real;
s:string(100);

{//////////}
{// hier werden die vom programm benoe- //}
{// tigten daten (materialconstanten, //}
{// order ?) eingelesen. //}
{// + wenige initialisierungen //}
{// + Daten fuer die Spektren //}
{// -----> main //}
{//////////}

begin
  reset(spe);
  readln(spe,histogramme);
  if histogramme>0 then begin
    readln(spe);
    for i:=1 to histogramme do begin
      readln(spe,titel[i]);
      read(spe,xmin[i],xmax[i],xbins[i]); readln(spe);
      read(spe,ymin[i],ymax[i],ybins[i]); readln(spe);
      readln(spe);
    end;
  end
  else begin
    histogramme:=1;
    titel[1]:='Default';
    xmin[1]:=0; xmax[1]:= 1; xbins[1]:= 1;
    ymin[1]:=0; ymax[1]:= 1; ybins[1]:=100;
  end;
  close(spe);
  qn:=0;
  reset(fil);
  readln(fil); readln(fil,s); readln(fil,s); readln(fil);
  readln(fil,paint,w1,w2,w3);
  readln(fil,NrofPhotons,NrBody);
  readln(fil,GrenzWahrscheinlichkeit,Surfaceloss);
  readln(fil,location);
  if location=1 then readln(fil,posx,dx,posy,dy);
  readln(fil);
  orde[1,1]:=1; orde[2,1]:=2; orde[3,1]:=3; orde[4,1]:=4; orde[5,1]:=1;
  orde[1,2]:=5; orde[2,2]:=1; orde[3,2]:=4; orde[4,2]:=8; orde[5,2]:=5;
  orde[1,3]:=8; orde[2,3]:=4; orde[3,3]:=3; orde[4,3]:=7; orde[5,3]:=8;
  orde[1,4]:=7; orde[2,4]:=3; orde[3,4]:=2; orde[4,4]:=6; orde[5,4]:=7;
  orde[1,5]:=6; orde[2,5]:=2; orde[3,5]:=1; orde[4,5]:=5; orde[5,5]:=6;
  orde[1,6]:=5; orde[2,6]:=6; orde[3,6]:=7; orde[4,6]:=8; orde[5,6]:=5;
  for number:=1 to NrBody do begin
    readln(fil,shape[number]);
    case shape[number] of
      quader: begin
        qn:=qn+1;
        rang[quader,number]:=qn;
        readln(fil,index[quader,qn]);
        readln(fil,interfaces);
        if interfaces>0 then begin
          for i:=1 to interfaces do begin
            read(fil,side);
            read(fil,interf[quader,qn,side]);
          end;
        end;
      end;
    end;
  end;
end

```

```

        end;
        readln(fil);
    end;
    for i:=1 to 8 do
        for k:=x to z do read(fil,corn[i,k]);
        readln(fil);
        for i:=1 to 6 do begin
            read(fil,mode[qn,i]);
            for j:=0 to 4 do begin
                for k:=x to z do id[qn,i,j,k]:=corn[orde[j+1,i],k];
                betrag_id[qn,i,j]:=square(id[qn,i,j]);
            end;
            for j:=x to z do
                pkt[j]:=(id[qn,i,3,j]-id[qn,i,1,j])/2+id[qn,i,1,j];
            plain[qn,i]:=area(pkt,qn,i);
        end;
        readln(fil); readln(fil);
    end;
    {//////////}
    {\ input quader }
    {//////////}
{.....}
    spiral: begin
        rang[spiral,number]:=1;
        readln(fil,mode_stirn);
        readln(fil,index[spiral,1]);
        readln(fil,rho_d,seg,rad_0,zmax);
        readln(fil);
        interf[spiral,1,1]:=2;
        zmin:=0;
        rho_d:=rho_d*fac;
        seg:=seg*fac;
        if (rho_d>pi+0.2) then rho_0:=pi-rho_d-seg/2
        else rho_0:=-seg/2;
        mrad:=2;
        end;
        {//////////}
        {\ input spiral }
        {//////////}
{.....}
    end; {---case---}
end;

{//////////}
{/ folgender abschnitt bestimmt die /}
{/ koordinaten eines gerade ueber den quader /}
{/ gestuelpten wuerfels. /}
{/ ---> in verbindung mit point_quader zum /}
{/ generieren von photonen im quader /}
{/ gebraucht. /}
{//////////}
{if shape[1]=quader then begin}
    for koor:=x to z do begin
        mi[koor]:=10000;
        ma[koor]:=-10000;
    end;
    for i:=1 to 6 do
        for j:=1 to 4 do
            for koor:=x to z do begin
                if id[1,i,j,koor]<mi[koor] then mi[koor]:=id[1,i,j,koor];
                if id[1,i,j,koor]>ma[koor] then ma[koor]:=id[1,i,j,koor];
            end;
        end;
    end;
{end;}
end;

```

```

{===== input =====}
{===== transmission =====}
function fresnel:real;
{////////////////////////////////}
var xw:real;
    koor,NewBody,NewSubBody:integer;
    n1,n2,hy,an,ge, faktor,erg, bn,bf:real;
    proj,anti,lot,hyp,ank,gek, new:vector;
    fpara,fortho,para1,para2 : vector;
    sina,sinb,cosa,cosb ,ro,rp ,fo,fp : real;
    sinab_p,sinab_m,cosab_p,cosab_m:real;

    {////////////////////////////////}
    {/ berechnet je nach sache die reflexions- /}
    {/ oder transmissions- wahrscheinlichkeit. /}
    {/ es werden keine trigonometrischen fktnen /}
    {/ verwendet (!) /}
    {/ -----> traverse /}
    {////////////////////////////////}

begin
    n1:=index[ shape[body],rang[shape[body],body] ];
    case PROPERTY of
        finit : n2:=nmedium;
        passiv: n2:=nluft;
        activ : begin
            NewBody:=interf[shape[body],rang[shape[body],body],side];
            n2:=index[shape[NewBody],rang[shape[NewBody],NewBody]];
            end;
    end;

    {////////////////////////////////}
    {/ waehle brechungsindizes /}
    {.....}

    ortho(plane[2],plane[3],lot);
    t:=factor(point,lot,plane);
    for koor:=x to z do begin
        proj[koor]:=point[koor]+t*lot[koor];
        ank[koor]:=(proj[koor]-point[koor]);
        gek[koor]:=(refl[koor]-proj[koor]);
        hyp[koor]:=(refl[koor]-point[koor]);
    end;
    an:=amount(ank);
    ge:=amount(gek);
    hy:=amount(hyp);
    if (hy<1E-14)or(an<1E-14)or(ge<1E-14)or(hy<ge)or(hy<an) then begin
        fresnel:=0;
        writeln('photon:',photon,' ----murks-in-geometry---');
        end
    else begin
    {-----}
        for koor:=x to z do begin
            para1[koor]:=scalar(polaris,gek)/sqr(ge)*gek[koor];
            para2[koor]:=scalar(polaris,ank)/sqr(an)*ank[koor];
            fpara[koor]:=para1[koor]+para2[koor];
            fortho[koor]:=polaris[koor]-fpara[koor];
        end;
        sina:=ge/hy;
        if sina < n2/n1 then begin { -----> }
            if sina<null then sina:=null;
            total:=false;
        end;
        cosa:=an/hy;
    end;
end;

```

```

sinb:=sina*n1/n2;
cosb:=sqrt(1-sqr(sinb));
sinab_p:=sina*cosb+sinb*cosa;
sinab_m:=sina*cosb-sinb*cosa;
cosab_p:=cosa*cosb-sina*sinb;
cosab_m:=cosa*cosb+sina*sinb;
ro:=abs(sinab_m/sinab_p);
rp:= (sinab_m/cosab_m/sinab_p*cosab_p);
if PROPERTY in [activ,finit] then begin
  rp:=sqrt(1-sqr(rp)); ro:=sqrt(1-sqr(ro));
  faktor:=cosb/cosa * n2/n1;
  for koor:=x to z do begin
    aim[koor]:=refl[koor]-(proj[koor]+faktor*(point[koor]-proj[koor]));
    point[koor]:=refl[koor];
  end;
  {----- neue richtung! --}
  new[x]:=fortho[y]*aim[z]-fortho[z]*aim[y];
  new[y]:=fortho[z]*aim[x]-fortho[x]*aim[z];
  new[z]:=fortho[x]*aim[y]-fortho[y]*aim[x];
  bn:=amount(new); bf:=amount(fpara);
  for koor:=x to z do
    polaris[koor]:= ro*fortho[koor] + rp*new[koor]/bn*bf;
  {-----}
  {polaris[koor]:= ro*fortho[koor] + rp*(para1[koor]+para2[koor]);}
end
else for koor:=x to z do begin
  aim[koor]:=refl[koor]-(point[koor]+2*t*lot[koor]);
  point[koor]:=refl[koor];
  polaris[koor]:= ro*fortho[koor] + rp*(para2[koor]-para1[koor]);
end;
{ ----- }

end
else begin
  total:=true;
  if PROPERTY in [finit,activ] then begin
    for koor:=x to z do begin
      point[koor]:=refl[koor];
      polaris[koor]:=0;
    end;
  end
  else for koor:=x to z do begin
    aim[koor]:=refl[koor]-(point[koor]+2*t*lot[koor]);
    point[koor]:=refl[koor];
    polaris[koor]:= fortho[koor] + (para2[koor]-para1[koor]);
  end;
  { <----- }

end;
fresnel:=square(polaris);
end;
end;
{.....}
procedure traverse(nr,shape:integer);
{//////////}
var koor:integer;
begin
  {----- traverse -----}
  { reflectiert ( --> reflect ) solange, bis }
  { strahl verloren geht oder 'exit'-flaeche }
  { getroffen wird. }

```

```

                                { output                point,aim : vec      }
                                {-----}
REPEAT
  Reflections:=Reflections+1;
  case shape of
    quader: point_quader(nr);
    spiral: point_spirale;
  end;
  PROBABILITY:=fresnel;
  points[Reflections,x]:=fktx(point);
  points[Reflections,y]:=fktz(point);
  attenuate:=Attenuated(PROBABILITY);
  if (attenuate < GrenzWahrscheinlichkeit) or outside
                                then REFLECTED:=false;
  if outside then begin
    writeln('murks ', photon);
    murks:=murks+1;
  end;
  UNTIL not REFLECTED or (PROPERTY in [activ,finit]);
  outside:=false;
end;
{===== traverse =====}
{===== main =====}
begin
  pi:=arctan(1)*4; fac:=pi/180;
  paint:=0; murks:=0; tube:=0;
  termont(outp);
  datetime(date,tim);
  times:=clock/1000000;
  writeln(outp,date,' ',tim);
  input;
  MINx:=-rad_0*cos(seg/2);
  head_spirale;
  view;
  order_init;
  {\ ----> \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ loop \\\}
  for photon:=1 to NrofPhotons+murks do begin
    PROBABILITY:=1; REFLECTED:=true; way:=0; Reflections:=0; loss:=1;
    total:=true; body:=1;
    if location=1 then Generate_Photon else
      case shape[1] of
        quader: Generate_Quader;
        spiral: if rndm(nix) > 0.9494 then begin
                                Generate_Quader;
                                body:=2;
                              end
              else Generate_Spirale;
      end;
    points[Reflections,x]:=fktx(point);
    points[Reflections,y]:=fktz(point);
    Generate_Polarisation;
                                {\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\}
                                {\ generieren \\\}
  {.....\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\}
    for koor:=x to z do old_p[koor]:=point[koor];
    for koor:=x to z do origin[koor]:=point[koor];
    PROPERTY:=passiv;
  {.....}
  REPEAT

```



```

    case shape[body] of
      quader: SubBody:=rang[quader,body];
      spiral: begin
        side:=1;
        SubBody:=1;
        end;
    end;
  traverse(SubBody,shape[body]);
  if PROPERTY=activ then body:=interface(shape[body],SubBody,side);
  UNTIL (PROPERTY=finit) or not REFLECTED;
  {\\ d\\ durchqueren \\}
{.....\\}
  if paint>0 then begin
    for body:=1 to NrBody do
      case shape[body] of
        quader: begin
          SubBody:=rang[shape[body],body];
          graphic_quader(SubBody);
          end;
        spiral: graphic_spirale;
        end;
      move;
      grnxtf;
      end;
      {\\ zeichnen \\}
{.....\\}
      if (PROPERTY=finit) and REFLECTED then sort; {\\ sortieren \\}
{.....\\}
      scale:=1000;
      if abs(photon-trunc(photon/scale)*scale)<0.001 then begin
        writeln(outp,photon:6,' * Events time: ',clock/1000000-times:5:5);
        end;
      end;
    {\\ loop \\} <----- \\}
    datetime(date,tim);
    writeln(outp,tim);
    order_print(histogramme);
  END;
  {\\ OPAQUE \\}
  ...

```

E Zusätzliche Ergebnisse der Messung

Die folgenden 3 Abbildungen zeigen die an den Bestrahlungsorten A_i , C_i und D_i der ersten Konfiguration aufgenommenen Lichtausbeuten (E_i und B_i ; siehe 3.3.2). Dabei sind die Mittelwerte der Einzelspektren gegen die Position der Photomultiplier auf der 'Quirl'-Stirnfläche aufgetragen⁶.

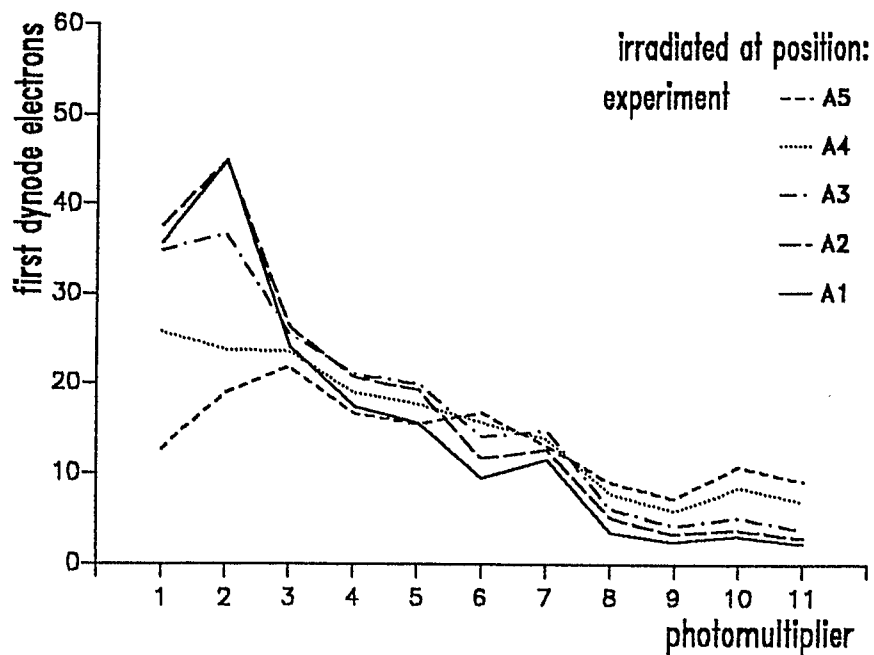
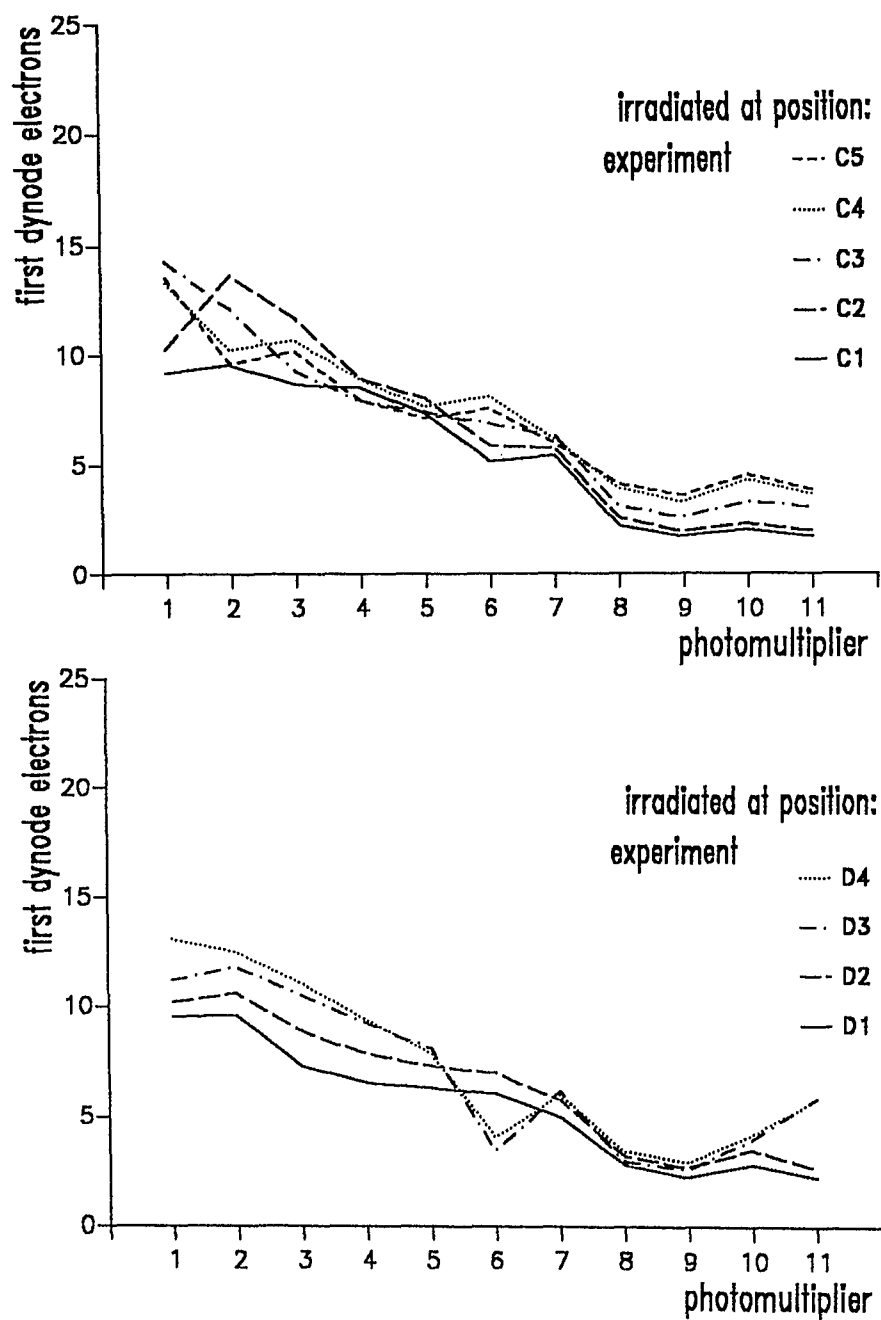


Abbildung 5.6: Aufgetragen ist die Anzahl der im Mittel pro Photomultiplier registrierten Photonen, gegen die Position der Röhren auf der Stirnfläche (hier für die Bestrahlungsorte A_i).

⁶Die am besten geeignete Darstellungsform wäre das Histogramm. Um aber 3 bis 5 Messungen in einem Diagramm wiedergeben zu können, wurden die diskreten Mittelwerte durch Kurven miteinander verbunden.

Abbildung 5.7: Bestrahlungsorte C_i (oben) und Bestrahlungsorte D_i (unten).

Zu den Bestrahlungsorten E_i - A_i folgen hier noch die Laufzeiten der Photonen bis zur ihrem Nachweis im Photomultiplier, aufgetragen gegen dessen Position auf der Stirnfläche des 'Quirls' (Konfiguration 1). Dargestellt sind die aus den Spektren bestimmten Positionen der Maxima und ihre Halbwertsbreite.

Hingewiesen sei auf die unterschiedlichen (relativen) Zeitskalen. Der Zeitnullpunkt liegt bei $t_0 = 26.33 \text{ nsec}$.

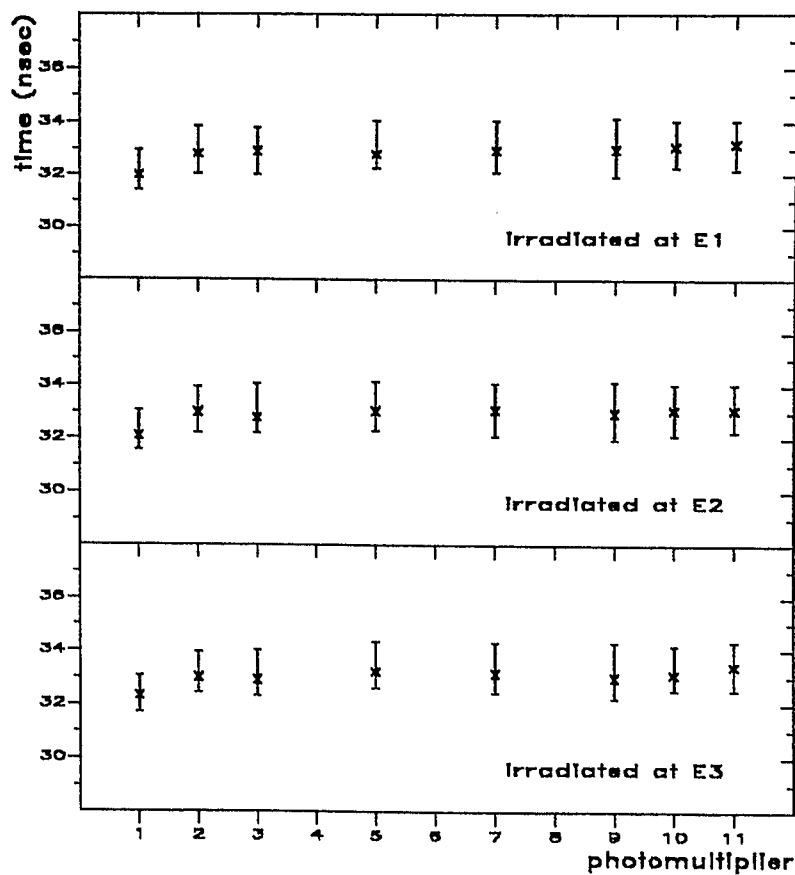


Abbildung 5.8: Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (E_i).

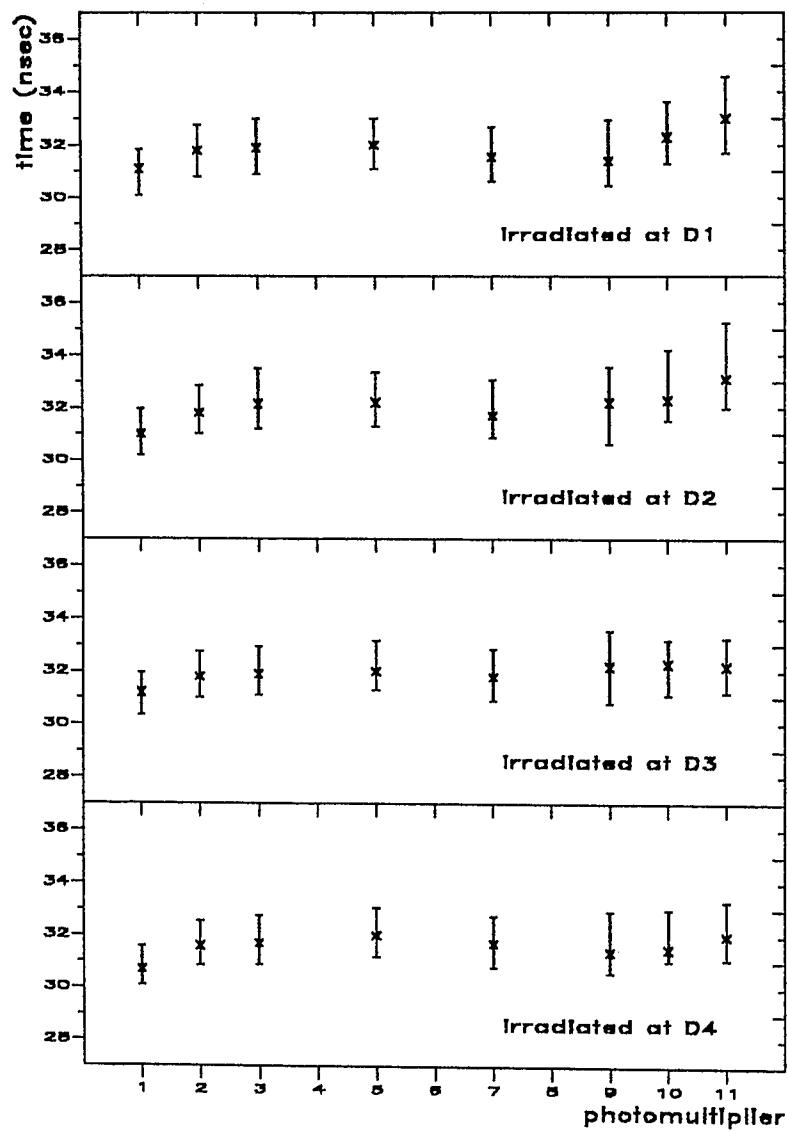


Abbildung 5.9: Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (D_i).

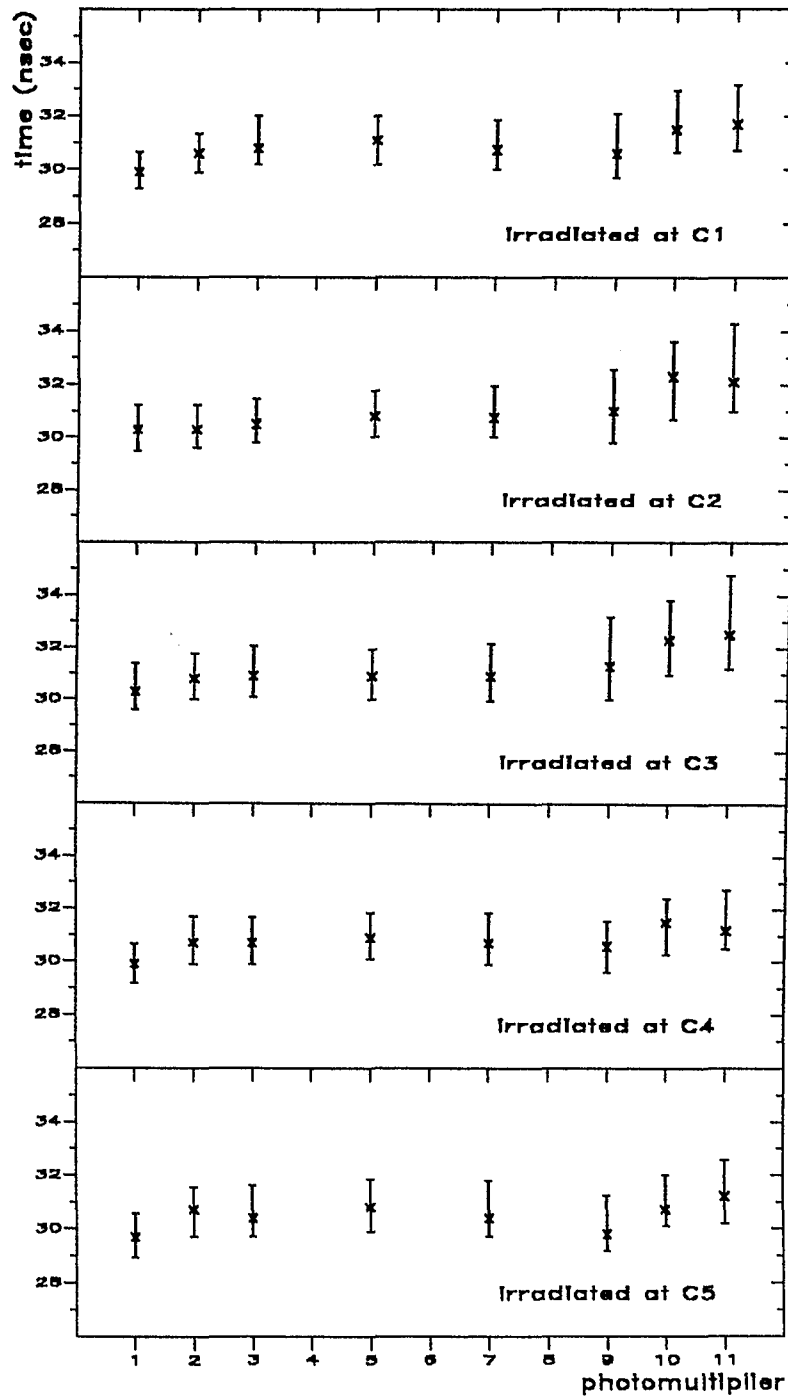


Abbildung 5.10: Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (C_i).

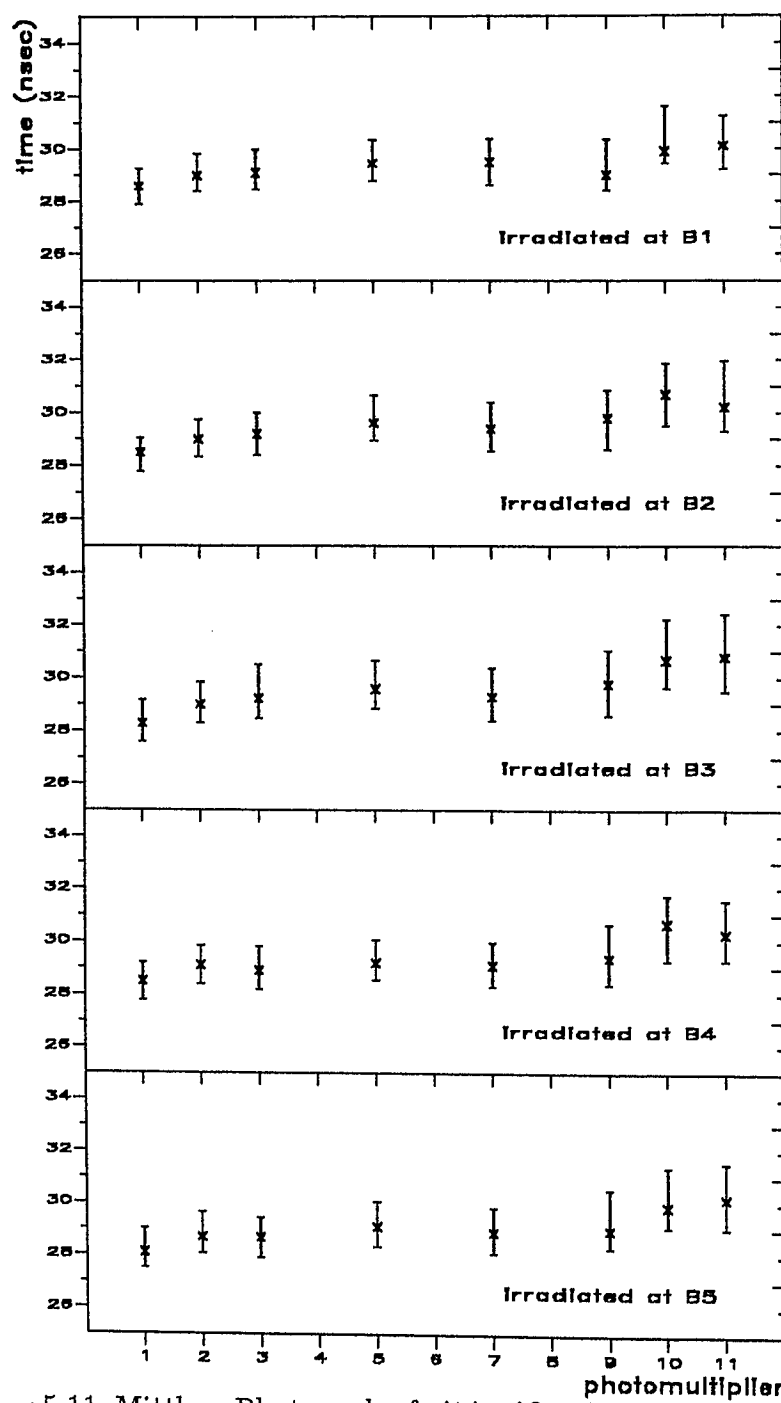


Abbildung 5.11: Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (B_i).

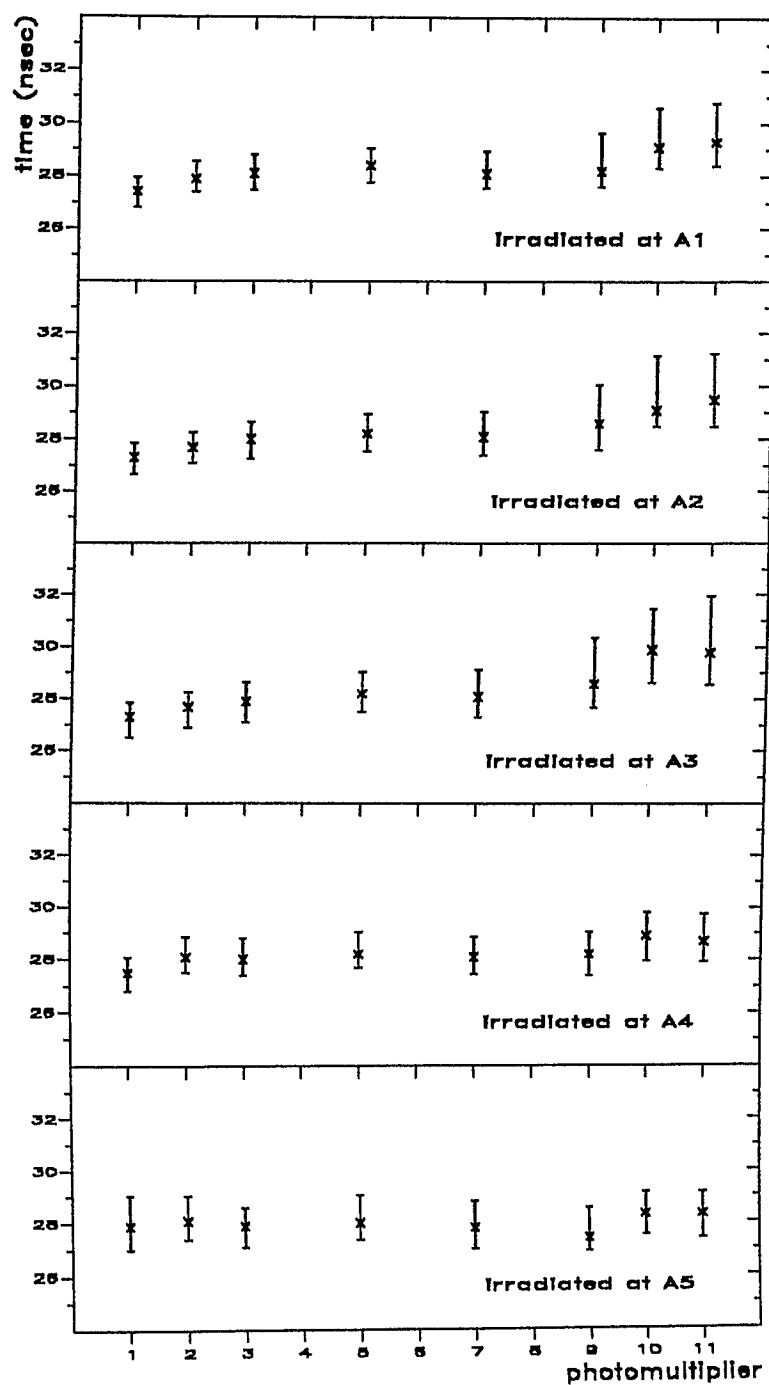


Abbildung 5.12: Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (A_i).

Abbildungsverzeichnis

1.1	Quarklinienbild zu der Zweig unterdrückten Reaktion $\bar{p}p \rightarrow \Phi\Phi$	3
1.2	Das kompakte Detektorsystem JETSET	4
1.3	Totale Wirkungsquerschnitte für die in der ersten Phase zu untersuchende Reaktion $\bar{p}p \rightarrow K^+K^-K^+K^-$ (rechts unten) und die dominanten Untergrundreaktionen ($\bar{p}p \rightarrow X$). [Rep88]	5
1.4	Gedrehte Szintillatorlage der Endkappe	6
2.1	Flußdiagramm für das Programm OPAQUE.	10
2.2	Projektion der durch ECKE DATA erzeugten Anordnung von Lichtleitern. Der Punkt rechts zeigt den Ort der Generation, die Stirnfläche von Quader 4 dient der Lichtauskopplung.	13
2.3	Konstruktion der neuen Photonrichtung. Für die Reflexion erhält man sie durch spiegeln von <i>point</i> an der Reflexionsebene $\rightarrow aim(r)$; im Falle der Transmission über das Verhältnis $An'/An \rightarrow aim(t)$	18
2.4	Projektion der Geometrie des 'Quirl'-segments ohne den angesetzten Quader ($xMin = -46.36cm$).	19
2.5	Projektion der im Datenfile QUIRL definierten Lichtleiterkombination ('Quirl' + angesetzter Quader).	20
2.6	Dargestellt ist die Stirnfläche mit den angesetzten Plexiglaslichtleitern.	24
2.7	Darstellung der Effizienz eines 'Quirl'-segments.	25
2.8	Projektion der homogen bestrichenen Einheitskugel-Oberfläche. Die Punktdichte nimmt vom Zentrum nach außen hin sinusförmig zu.	26
2.9	Winkelverteilung der Photonen, die die Stirnfläche des 'Quirls' erreichen.	27
2.10	Luminosität der 'Quirl'- Stirnfläche.	28
2.11	Zeitspektren, simuliert mit an den Punkten A3, D1 und E2 lokal generierten Photonen.	29
2.12	Simulierte Lichtausbeute der Photomultiplier aufgetragen gegen deren Position (normiert auf 7000 generierte Photonen).	30

2.13	Simulierte integrale Lichtausbeute aufgetragen gegen die Bestrahlungsorte quer zum 'Quirl' (normiert auf 7000 generierte Photonen, siehe auch Abbildung 3.22).	31
2.14	'Totalreflektions-Kegel': alle Photonen, deren Spur innerhalb des gezeigten Kegels liegt, werden nicht an der Ebene E reflektiert.	32
2.15	Durch Seitenfläche abgeschnittener 'Totalreflexionskegel'. Die Wahrscheinlichkeit für eine Reflexion an der Bodenfläche ist davon unabhängig.	33
2.16	Winkelverteilung der Photonen an der Auskopplungsfläche eines rechteckigen Stabes. Dargestellt analog zur Winkelverteilung des 'Quirls'.	34
3.1	Schema des Versuchsaufbau mit vereinfacht dargestellter Stirnfläche	38
3.2	BC 404 Emissionsspektrum [BIC74]	40
3.3	(a) Skizze der Flächen, an die Photomultiplier angesetzt wurden mit einem Beispiel für Konfiguration α_1 . (b) Generelle Form der Plexiglas-Ansatzstücke	41
3.4	Fertiger Aufbau der 2. Konfiguration	42
3.5	Schema einer Photomultiplerröhre [HAM88]	43
3.6	Spektrale Empfindlichkeit einer Bialkali Photokathode - typisch für die verwendeten Photomultiplier ¹⁵ . [HAM88]	45
3.7	Photomultiplier HAMAMATSU R1635-02	46
3.8	Blockschaltbild der Elektronik im Triggerzweig	47
3.9	Blockschaltbild der Elektronik im Signalzweig	48
3.10	Foto und schematische Zeichnung der gesamten Triggeranordnung . .	49
3.11	(a) Betaspektrum für Strontium/Yttrium (b) Reichweite von Elektronen in Szintillatormaterial	51
3.12	Vergleich zwischen Energieverteilung der vom Trigger selektierten Elektronen und Betaspektrum (Strontium/Yttrium)	52
3.13	Wahrscheinlichster Enerverlust für Elektronen in Szintillatormaterial aufgetragen gegen ihre kinetische Energie (Bethe-Bloch).	53
3.14	Abhängigkeit des 'stragglings' der Elektronen von ihrer kinetischen Energie.	54
3.15	Das Photo zeigt den gesamten Versuchsaufbau in der schwarz ausgekleideten Aluminiumkiste (ohne Trigger). Links oben, an der Spitze des 'Quirls', ist die Leuchtdiode zu sehen, die zum Eichen der Photomultiplier benutzt wurde.	55

3.16	(a) Spektrum der Referenzsignale - die Halbwertsbreite beträgt etwa 3 Kanäle; (b) Eichgerade deren reziproke Steigung proportional der eingestellten Verstärkung ist.	57
3.17	Dargestellt sind 3 der zum Eichen der Photomultiplier aufgenommenen 'single electron peaks'.	58
3.18	Vergleich dreier geeichter Spektren: PMT1, PMT5 und PMT10 - Bestrahlungsort E2, 100 000 registrierte Ereignisse.	60
3.19	Mittlere Anzahl der registrierten Photonen pro Ereignis, aufgetragen gegen die Photomultiplierposition auf der Stirnfläche - Bestrahlungsorte E _i	61
3.20	Mittlere registrierte Photonenzahl pro Ereignis aufgetragen gegen die Multiplierposition - Bestrahlungsorte B _i	62
3.21	Integrale Lichtausbeute des 'Quirls': Bestrahlungsort E2, summiert über die 11 Photomultiplier.	63
3.22	Mittelwerte der integralen Lichtausbeute des 'Quirls' pro Ereignis für alle Meßpositionen: Konfiguration 1.	64
3.23	Mittelwerte der integralen Lichtausbeute des 'Quirls' pro Ereignis für alle Meßpositionen: Konfiguration 2. (Die Positionen 1 - 5 liegen quer zum 'Quirl', A - E entlang dem 'Quirl'.)	65
3.24	Vergleich der Zeitspektren dreier Photomultiplier, Bestrahlungsort: C1. Die Zeitskala besitzt einen relativen Zeitnullpunkt.	66
3.25	Mittlere Laufzeit der Photonen im 'Quirl', aufgetragen gegen die Photomultiplierpositionen.	67
3.26	Weg-Laufzeit-Diagramm für die Meßpunkte, die auf der 'Mittelspirale' liegen (A ₃ /B ₃ /C ₃ /D ₂ /E ₂).	68
3.27	Zusammenhang zwischen Zeitaufösung und mittlerer Anzahl registrierter Photonen \bar{n} für PMT1.	69
5.1	Testspektren dreier Photomultiplier R1635-02	77
5.2	Base E1761-04 mit Schaltbild [HAM86]	78
5.3	(a) Hauptmenü,(b) das zu 'P'arameters gehörige Menü mit welchen die Spektren beeinflußt werden.	88
5.4	Gestalt des Bildschirms während einer Messung.	94
5.5	EinzeIspektrum, mit der Möglichkeit durch erneutes betätigen der Taste P eine Hardcopy zu erzeugen.	95
5.6	Aufgetragen ist die Anzahl der im Mittel pro Photomultiplier registrierten Photonen, gegen die Position der Röhren auf der Stirnfläche (hier für die Bestrahlungsorte A _i).	119
5.7	Bestrahlungsorte C _i (oben) und Bestrahlungsorte D _i (unten).	120

5.8	Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (E_i).	121
5.9	Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (D_i).	122
5.10	Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (C_i).	123
5.11	Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (B_i).	124
5.12	Mittlere Photonenlaufzeit im 'Quirl' aufgetragen gegen die Position des Photomultipliers (A_i).	125

Bibliographie

- [Alo70] Alonso-Finn. *Physics*. Addison-Wesley Publishing Company, Inc., 1970.
- [Bet30] H. A. Bethe. *Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie*. Ann. Physik, 16:285, 1930.
- [BIC74] BICRON. *Premium Plastic and Liquid Scintillator*. 1974.
- [BIC86] BICRON. *physical constants of Plastic Scintillators*. August 1986.
- [Bir53] J. B. Birks. *Scintillation counters*. McGraw-Hill Book Company, Inc., New York, 1953.
- [Bir67] J. B. Birks. *The Theory and Practice of Scintillation Counting*. Pergamon Press, Department of Physics University of Manchester, 1967.
- [BS84] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*, page 94. Verlag Harry Deutsch Thun und Frankfurt (Main), 1984.
- [Fer86] Richard C. Fernow. *Introduction to experimental particle physics*. Cambridge University Press, Brookhaven National Laboratory, 1986.
- [Gro88] Particle Data Group. *Particle Properties Data Booklet*. from 'Review of Particle Properties' Physical Letters 170B, North Holland, Amsterdam, April 1988.
- [HAM86] HAMAMATSU. *Accessories For Photomultiplier Tubes*. Japan, Juli 1986.
- [HAM88a] HAMAMATSU. *Final Test Sheet*. June 1988.
- [HAM88b] HAMAMATSU. *Photomultiplier Tubes*. Japan, Januar 1988.
- [Hur85] C. R. Hurlbut. *Plastic Scintillators - A Survey*. BICRON, November 1985.

- [Jac83] J. D. Jackson. *Klassische Elektrodynamik*. de Gruyter, 2. Auflage, S. 325 ff, 1983.
- [LC77] B. Leskovar and C.C.Lo. *Transit time spread measurements of microchannel-plate photomultipliers*. IEEE, September 1977.
- [LeC85] LeCroy. *Operator's Manual Modell 2228A, octal Time-to-Digital Converter*. Februar 1985.
- [LeC86] LeCroy. *ADC User's Manual Modell 4300B*. Februar 1986.
- [MY68] J. B. Marion and F. C. Young. *Nuclear Reaction Analysis (graphs and tables)*. North Holland Publishing Company, Amsterdam, Department of Physics and Astronomy University of Maryland, 1968.
- [Per82] D. H. Perkins. *Introduction to High Energy Physics*. Addison-Wesley Publishing Company, Inc., 1982.
- [PVTf87] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. *Numerical Recipes, the art of scientific computing*. Cambridge University Press, 1987.
- [RCA80] RCA. *Photomultiplier Handbook*. R. W. Engstrom, Lancaster, 1980.
- [Rep88] Cern Report. *The Jetset Experiment at LEAR, a status report to the PS CC*. november 1988.
- [Sef88] Thomas Sefzick. *Der schnelle Triggerszintillator für das JETSET-Experiment*. Diplomarbeit, Rheinische Friedrich-Willhelms Universität Bonn, Dezember 1988.
- [Tho83] L. Thornhill. *Scintillation Counter Technology*. CERN, Geneva, September 1983.
- [VAL70] VALVO. *Photomultipliers*. J. M. Schukeren, Hamburg, April 1970.

